

COMP.SE.100-EN ItSE

Zoom begins soon...
at 1415 o'clock.

HI, WHO JUST JOINED?	CAN YOU EMAIL THAT TO EVERYONE?	IS ____ ON THE CALL?	UH, ____ YOU'RE STILL SHARING...	HEY, GUYS, I HAVE TO JUMP TO ANOTHER CALL
(SOUND OF SOMEONE TYPING, POSSIBLY WITH A HAMMER)	(LOUD, PAINFUL ECHO/ FEEDBACK)	(CHILD OR ANIMAL NOISES)	HI, CAN YOU HEAR ME?	NO, IT'S STILL LOADING.
NEXT SLIDE, PLEASE.	CAN EVERYONE GO ON MUTE?	I'M SORRY; I WAS ON MUTE	(FOR OVERTALKERS) SORRY, GO AHEAD	HELLO? HELLO?
SO (cuts out) I CAN (unintelligible) BY (cuts out) OK?	SORRY I'M LATE (INSERT LAME EXCUSE.)	I HAVE A HARD STOP AT...	I'M SORRY, YOU CUT OUT THERE.	CAN WE TAKE THIS OFFLINE?
I'LL HAVE TO GET BACK TO YOU.	CAN EVERYONE SEE MY SCREEN?	SORRY, I WAS HAVING CONNECTION ISSUES.	I THINK THERE'S A LAG.	SORRY, I DIDN'T CATCH THAT. CAN YOU REPEAT?

CONFERENCE CALL BINGO

COMP.SE.100-EN, 2020, course schedule v6c (02.09.2020)

week	lectures	exam	weekly exercises	project assignment (exercise work)	week
35	L1: course basics		--- sign to WE groups ---	sign for project = grouping...	35
36	Project Assignment explained		WE1: intro to requirements	grouping, groups to Moodle	36
37	L2: Sw Eng in general		WE2: Trellis and agile way	group's Trello board ready with product backlog	37
38	L3: requirements		WE3: feasibility study and stakeholder analysis	working...	38
39	L4: basic UML diagrams		WE4: requirements	working...	39
40	L5: more UML diagrams	EXAM-1	WE5: UML diagrams - Use case	working...	40
41	L6: different sw systems	EXAM-1	WE6: UML diagrams - concept/entity and navigation	deadline for 1st phase documentation and presentation	41
42	examination week		examination week	examination week	42
43	L7: life cycle models		groups' 1st presentations	groups' 1st phase presentations	43
44	L8: quality and testing	EXAM-2	WE7: development processes	feedback group-to-group at PRP, from 1st phase	44
45	L9: project work	EXAM-2	WE8: testing and error reporting	deadline for diagrams first versions (Moodle)	45
46	L10: project management		WE9: effort estimation	feedback to groups from diagrams (from assistants)	46
47	L11: open source, APIs, IPR		WE10: delivery contracts and terms of use	deadline for 2nd phase presentation (PRP)	47
48	L12: embedded systems, IoT	EXAM-3	groups' final presentations	groups' final presentations / feedback g-to-g (PRP)	48
49	L13: recap, summary	EXAM-3	---	final (2.) delivery of project documentation	49
50	examination week		examination week	feedback inside group, student-to-student at PRP	50
51	examination week		examination week	end of game / game over.	51
	Lectures: Wed at 1415-16.		Weekly exercises:		
			Mon 0815-10	AUTUMN 2020 (1-2. periods)	
			Mon 1215-14	are remote/distant learning.	
			Tue 0815-10		
			Tue 1415-16		
			Wed 0815-10.		

FUTURE EVENTS AND STUDIES

From a spark to a flame. Y-kampus organizes a variety of events focusing on current business-related themes. Choose the ones that interest you most and join us to find inspiration and build a network of like-minded individuals.

NEWS

4.9.2020

International GISU-competition: three teams from Tampere to the Final!

1.7.2020

GISU-competition: five teams to Semi-Final

18.5.2020

Growth Programme 2020

STUDIES

C-LAB**22.10. - 17.12.2020**

EVENTS

Innovation Challenges**22.10. - 10.12.2020****C-LAB | European edition****2.11. - 04.12.2020****Innovation Challenges | European edition****2.11. - 04.12.2020**

[<https://intra.tuni.fi/en/content/news/18824>]

These guidelines include the common rules and instructions for using video in distance learning. It is recommended that teachers go through the list briefly with the students in the first meeting.

Instructions for the teacher

- It is recommended to briefly review the common rules and the instructions for distance learning in the first meeting.
- **Follow the safety instructions** given on the video conferencing platform, eg invitation and meeting settings.
- Send the students participation instructions **in advance** (eg in connection with a calendar invitation)
- It is important that all participants use a **background image** or blur the background.
- It is the teacher's responsibility to **react immediately** if a participant shares with others content that violates the guidelines.
- Tell everyone in advance if you plan to record a teaching session. Justify the need (eg other students can watch the session later).
- It is recommended that keeping the camera turned on during recording is not mandatory.

Instructions for students

- We recommend using **a profile picture**, which makes it easier to identify the speaker even without video.
- **Always** blur the background of a video. Avoid unnecessary visibility into the space where you are.
- If possible, stay in a space that others will not enter during the teaching sessions.
- **Tell your loved ones** in the same space that you are attending a distance learning session that may use live image.
- Make sure that no inappropriate or confidential material is visible in the **background**. If possible, look for a quiet place for the teaching session where the background is neutral.
- Mute additional audio sources in the distance learning situation.
- **Shut down the camera** if something irrelevant enters the camera's shooting range and during breaks.
- If you do not want to comment orally during video recording, use the chat window.

Strongly recommended studies

Finnish course

- **Puheviestintä ja neuvottelutaito**

is the same as English

- **Speech Communication and Negotiation Skills**

Every student should take this voluntary/optional course !

COMP.SE.100 -EN "ItSE"

Introduction to Software Engineering

2020, 1-2. periods

5 credit units

04-UML-ItSE-2020-v7

Tensu: remember to start Zoom
lecture recording, at 1415

Prefer course Moodle over SISU information.

Students are recommended to follow Moodle News/messages.

Current at course (w 39)

TUNI, Updated: 18.09.2020 14:04: Teaching in the autumn semester

Teaching will be primarily delivered remotely, as in the 1st period, for the duration of the autumn semester until 31 December 2020. Students will only attend essential classes (such as laboratory classes, small group teaching sessions, or student support sessions) in person in campus locations. The faculties, the Language Centre, Continuous Learning Services (such as professional development services) and the schools at TAMK may have a justified reason to put in place different arrangements for the delivery of teaching and may need to provide lecture-based classes, for example, to first-year students. Wear personal protective equipment when necessary. Lectures will be primarily delivered remotely.

22.09.2020

[<https://intra.tuni.fi/en/content/news/18824>]

Instructions for students

Consider these at ItSE 2020;
WE, and PA 1st and final presentations.

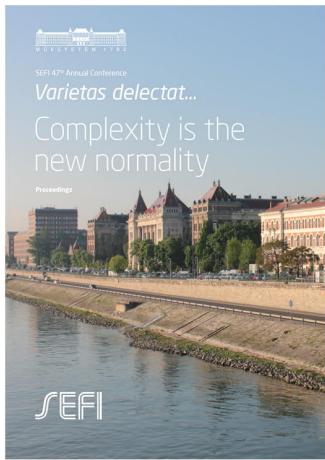
- We recommend using a **profile picture**, which makes it easier to identify the speaker even without video.
- **Always** blur the background of a video. Avoid unnecessary visibility into the space where you are.
- If possible, stay in a space that others will not enter during the teaching sessions.
- **Tell your loved ones** in the same space that you are attending a distance learning session that may use live image.
- Make sure that no inappropriate or confidential material is visible in the **background**. If possible, look for a quiet place for the teaching session where the background is neutral.
- Mute additional audio sources in the distance learning situation.
- **Shut down the camera** if something irrelevant enters the camera's shooting range and during breaks.
- If you do not want to comment orally during video recording, use the chat window.

TUNI / TIE-02301, 2019 (Scrumban and Trello)

Proceedings of the SEFI 47th Annual Conference · Concept Papers



Scrumban learning – agile, lean and transparent framework for practical learning experience



TUNI * COMP.SE.100-EN

Fig. 3 shows the results about how students viewed the usage of Kanban board in their practical work. Students could pick as many of the options as they wanted. Vast majority (90%) considered it helpful for visualizing the work.

Students felt that the iteration and dividing the work

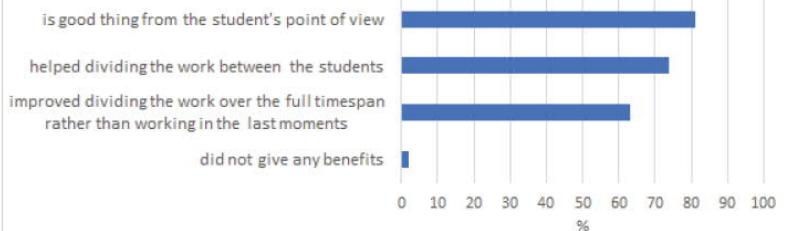
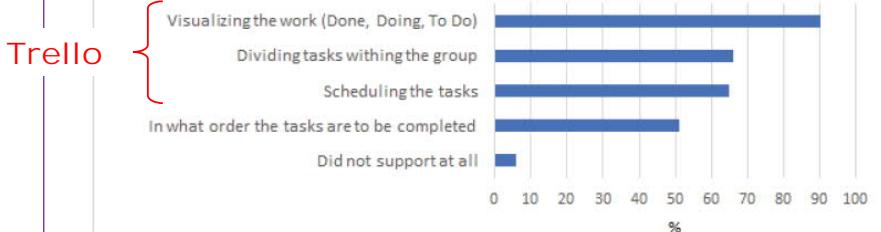


Fig. 2. Iteration and dividing of work

The Kanban -board supported the practical work with



23.09.2020 10.14 14

COMP.SE.100-EN, 2020, course schedule v6c (02.09.2020)

week	lectures	exam	weekly exercises	project assignment (exercise work)	week
35	L1: course basics		--- sign to WE groups ---	sign for project = grouping...	35
36	Project Assignment explained		WE1: intro to requirements	grouping, groups to Moodle	36
37	L2: Sw Eng in general		WE2: Trello and agile way	group's Trello board ready with product backlog	37
38	L3: requirements		WE3: feasibility study and stakeholder analysis	working...	38
39	L4: basic UML diagrams		WE4: requirements	working...	39
40	L5: more UML diagrams	EXAM-1	WE5: UML diagrams - Use case	working...	40
41	L6: different sw systems	EXAM-1	WE6: UML diagrams - concept/entity and navigation	deadline for 1st phase documentation and presentation	41
42	examination week		examination week	examination week	42
43	L7: life cycle models		groups' 1st presentations	groups' 1st phase presentations	43
44	L8: quality and testing	EXAM-2	WE7: development processes	feedback group-to-group at PRP, from 1st phase	44
45	L9: project work	EXAM-2	WE8: testing and error reporting	deadline for diagrams first versions (Moodle)	45
46	L10: project management		WE9: effort estimation	feedback to groups from diagrams (from assistants)	46
47	L11: open source, APIs, IPR		WE10: delivery contracts and terms of use	deadline for 2nd phase presentation (PRP)	47
48	L12: embedded systems, IoT	EXAM-3	groups' final presentations	groups' final presentations / feedback g-to-g (PRP)	48
49	L13: recap, summary	EXAM-3	---	final (2.) delivery of project documentation	49
50	examination week		examination week	feedback inside group, student-to-student at PRP	50
51	examination week		examination week	end of game / game over.	51
	Lectures: Wed at 1415-16.		Weekly exercises:		
			Mon 0815-10	AUTUMN 2020 (1-2. periods)	
			Mon 1215-14	are remote/distant learning.	
			Tue 0815-10		
			Tue 1415-16		
			Wed 0815-10.		

TUNI / COMP.SE.100-EN

23.09.2020 12.04

15

Course contents (plan)

1. Course basics, intro
2. Sw Eng in general, overview
3. Requirements
- 4. Basic UML Diagrams ("Class", Use Case, Navigation)**
5. UML diagrams, in more detail
6. Different software systems
7. Life Cycle models
8. Quality and Testing
9. Project work
10. Project management
11. Open source, APIs, IPR
12. Embedded systems
13. Recap

4. Basic UML Diagrams

- current course matters
- first some slides about Dia diagramming tool
- Entity diagram = context diagram, logical data model, "class diagram" (FI: käsitekaavio, tietoyhteykskaavio, käsitemalli), several versions exist depending of amount of information content terminogoly is not coherent
- Use Case diagram (FI: käyttötapauskaavio) and User Stories (FI: käyttäjatarinat)
- Navigation diagram (FI: navigointikaavio, navigatiokaavio) (not officially UML).

First, general course matters

Juanita: groups G01-G04

Aleksius: ODD groups; G05,G07,G09,G11,G13,G15,G17,G19,G21,G23,G27

Lauri: EVEN groups; G06,G08,G10,G12,G14,G16,G18,G20,G22,G24,G28

- Trello board and Product Backlog ready at the end of week 37
- invite (not add) your assistant to your Trello board.

WE attendees:

- Mon 0815-10 9, 8,10, 5,
- Mon 1215-14 11,12,12,13,
- Tue 0815-10 3, 6, 4, 6,
- Tue 1415-16 8,10, 9, 8,
- Wed 0815-10 12,11, 9, 8,

Current at course (w 39)

- WE4 were requirements
- EXAM 1/3 (w40-41) is open for reservations
- for WE5, install Dia (or EA) to your own computer
 - (EXAM classes have Dia and EA tools installed)
- all Trello boards, with group number, ready ??

short history of diagramming

Diagrams were introduced early in 1970s to help at requirements, when customers could not understand (technical) requirements specifications or did not care to read thick paper documents.

At that time there were no graphical displays, so pencil and paper were the tools.

In 1990s graphical diagramming software were becoming usual (but expensive !), and with mouse it was somehow easy to draw and modify diagrams.

At that time "every respected guru" designed his own diagramming style, and some of those methods were supported by a tool.

General and cheap drawing/diagramming tool was Windows Paint. But if you moved a box, the attached line did not move with. :-)

Nowadays technology has evolved at huge steps, with any diagramming tool you can draw many different kind of diagrams. There are also some "rules" at tools, so that you "must" follow some guidelines (automated method checking).

But you still have to think yourself what to draw. About some system, you can draw varying diagrams, so you have to decide how you model the system.

So, both text and diagrams are needed at requirements specification.

UML = Unified Modeling Language

This specification defines the Unified Modeling Language (UML), revision 2. The objective of UML is to provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of software-based systems as well as for modeling business and similar processes.

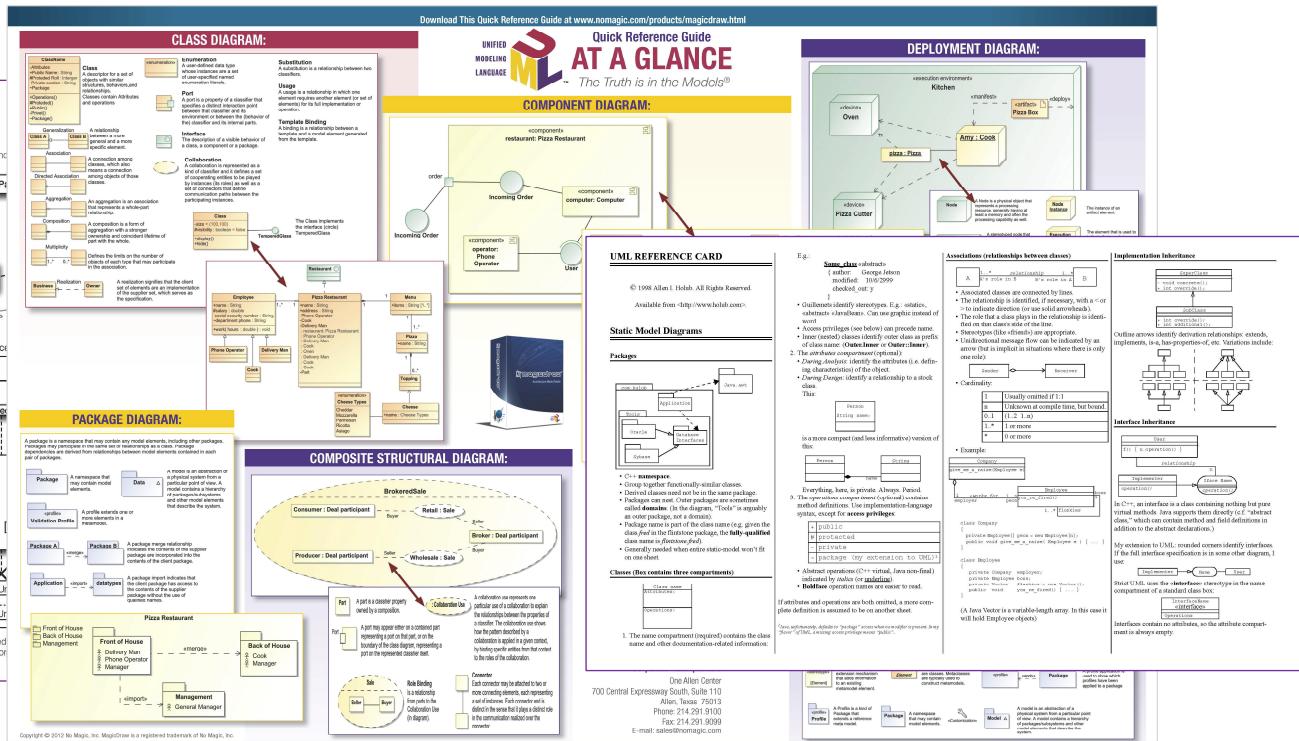
One of the primary goals of UML is to advance the state of the industry by enabling object visual modeling tool interoperability. However, to enable meaningful exchange of model information between tools, agreement on semantics and syntax is required.

Currently UML is the "industry standard" diagramming method, by OMG.

However, agree the diagramming style in your project, so that all parties understand it.



There are many UML reference cards and cheat sheets available



TUNI / COMP.SE.100-EN

22.09.2020 14.02

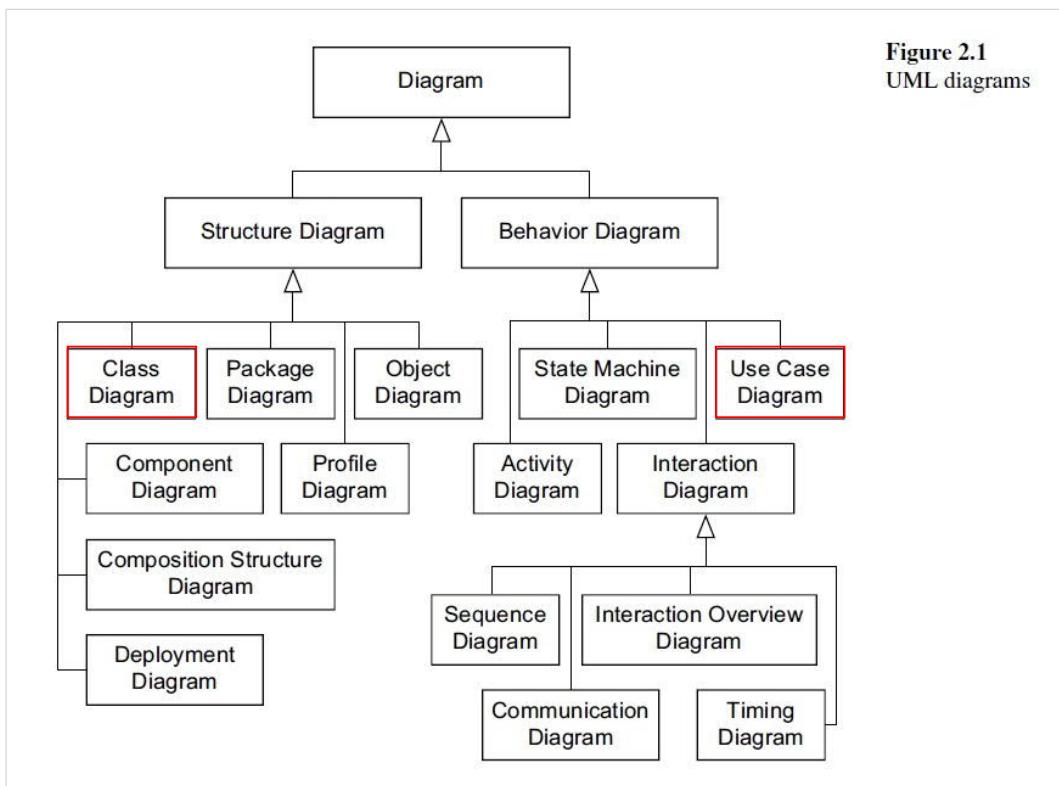
22

Three basic diagrams

- Entity diagram = context diagram (conceptual diagram, domain model, class diagram) shows entities and their connections/relations (associations) in the system. This does not reveal what are the functionalities, and in which order functionalities (should) happen. Usually drawn by tool's class diagram symbols.
 - Use Case diagram tells user groups and functionalities (actions). One functionality is explained in detail in textual User Story. This does not reveal what data moves in system and in which order functionalities (should/can) happen.
 - Navigation diagram (display map, menu hierarchy) figures out how user can navigate (move) inside the program. This does not reveal who (user group) does (should/can) what, and what data moves in the system.

All these are used in the requirements phase, first in general level and then updated with more detail. Navigation diagram would be useful later in User Guide/Manual.

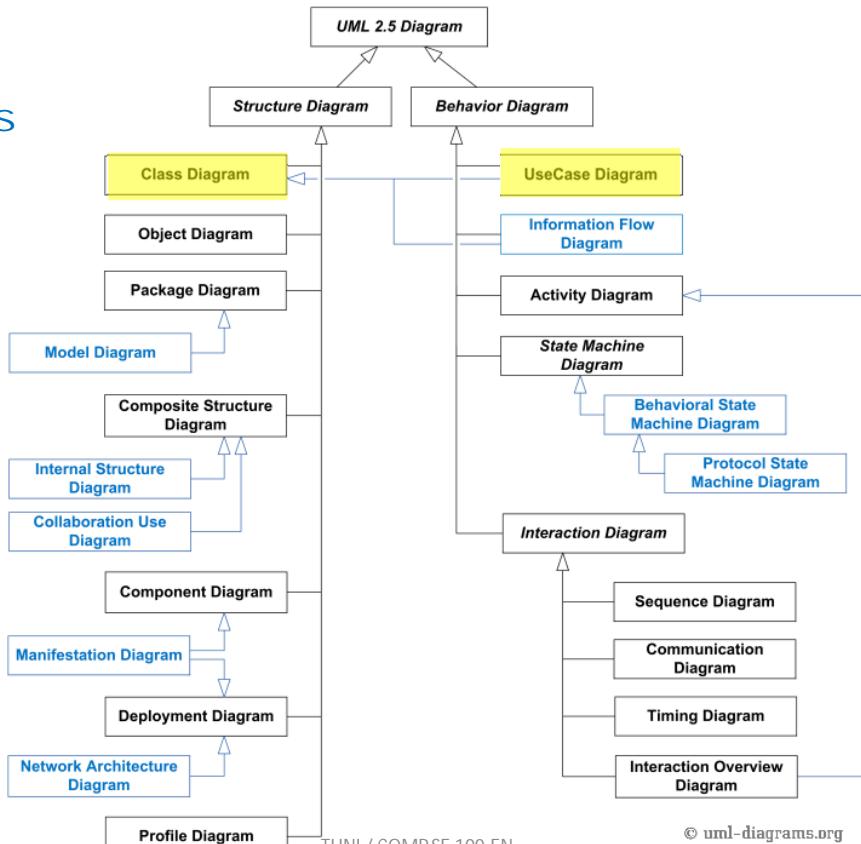
UML diagram types

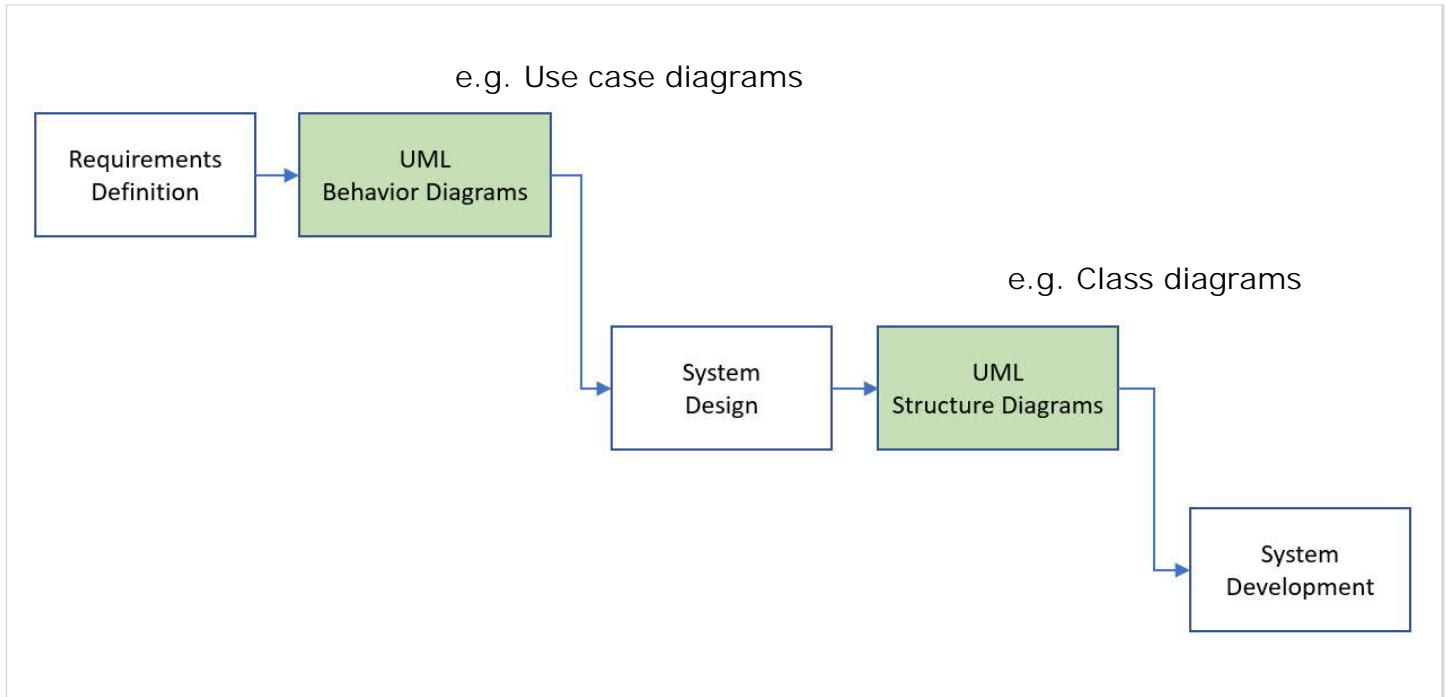


UML 2.5 diagrams (2015)

UML diagram types

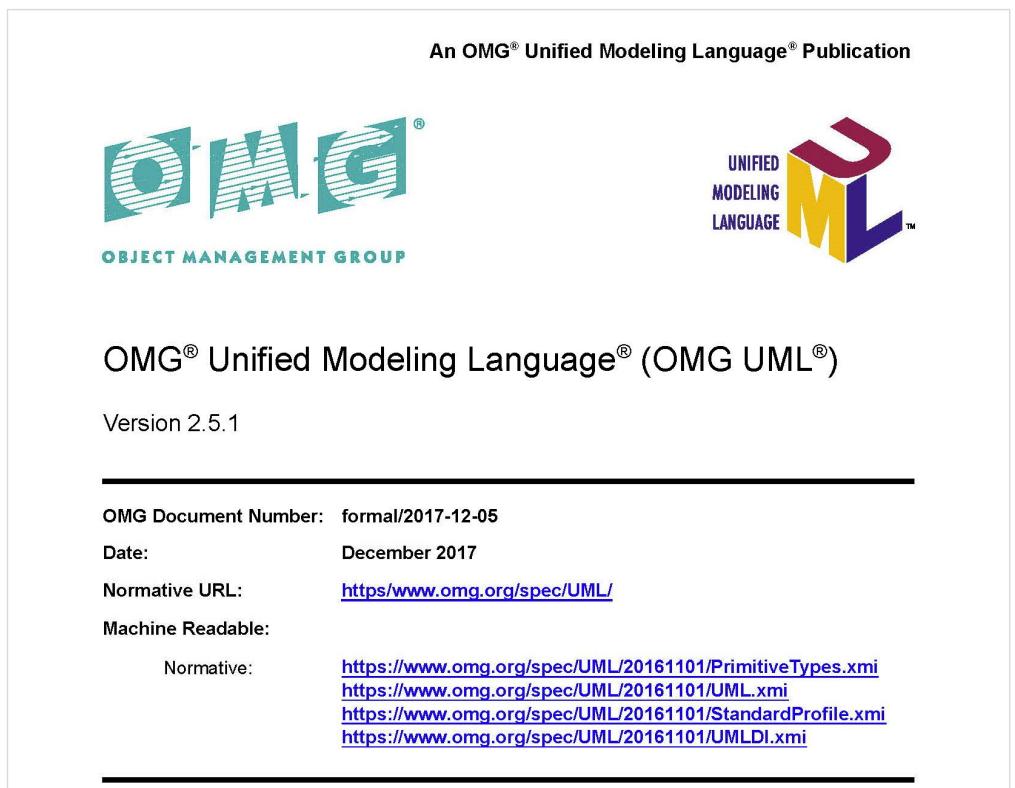
www.uml-diagrams.org/



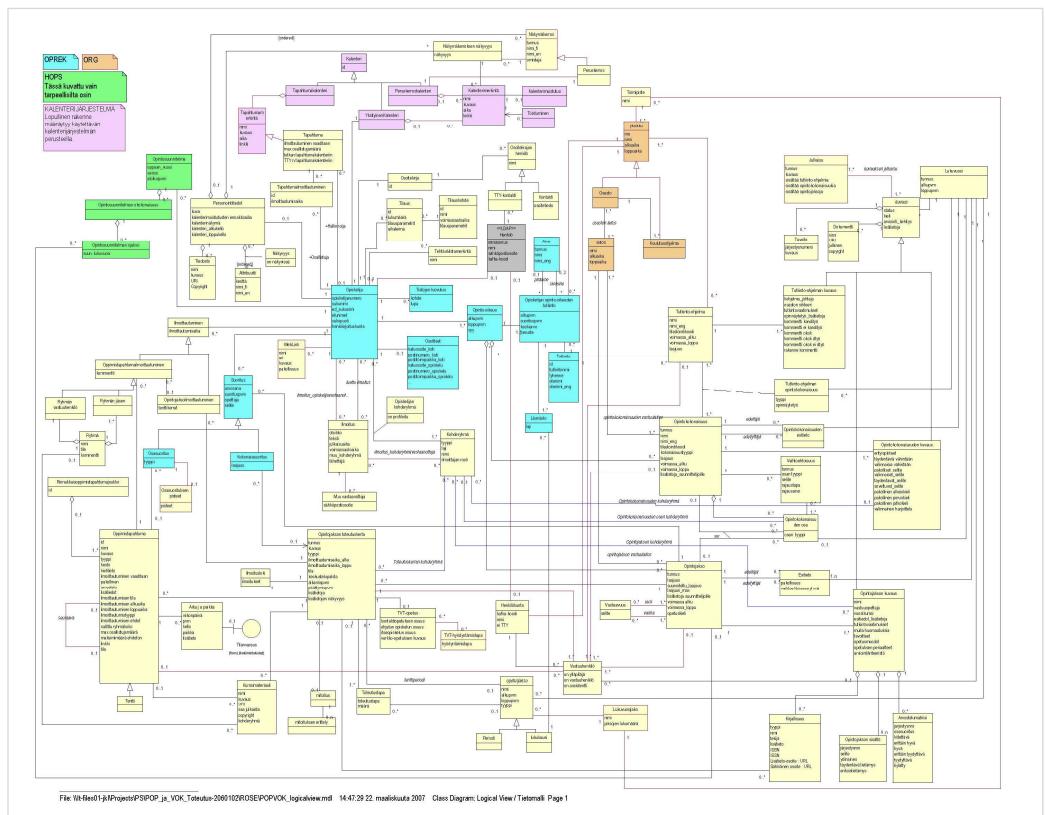


UML 2.5.1 (2017) specification

[www.omg.org/
spec/UML/](http://www.omg.org/spec/UML/)



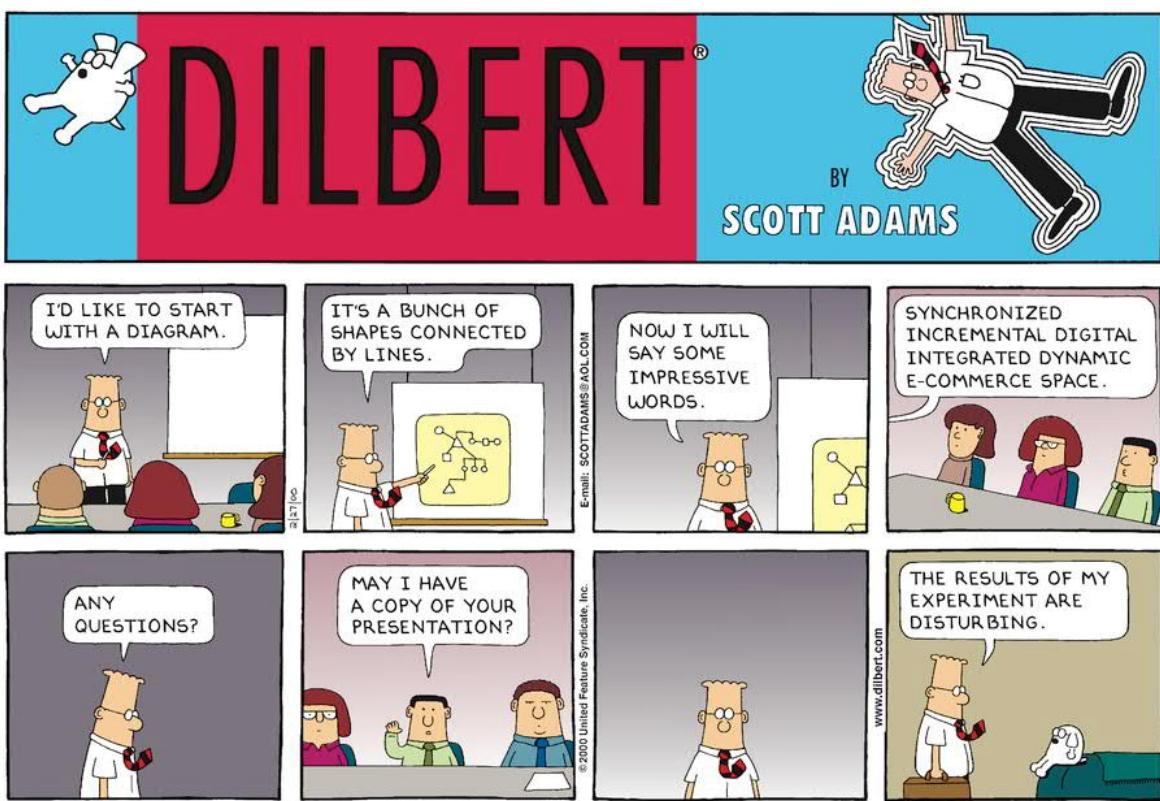
Too big
diagram ?
Here:
POP/ROCK.
Split diagram
or get larger
display.



TUNI / COMP.SE.100-EN

22 09 2020 14 02

28



first a few slides about Dia tool

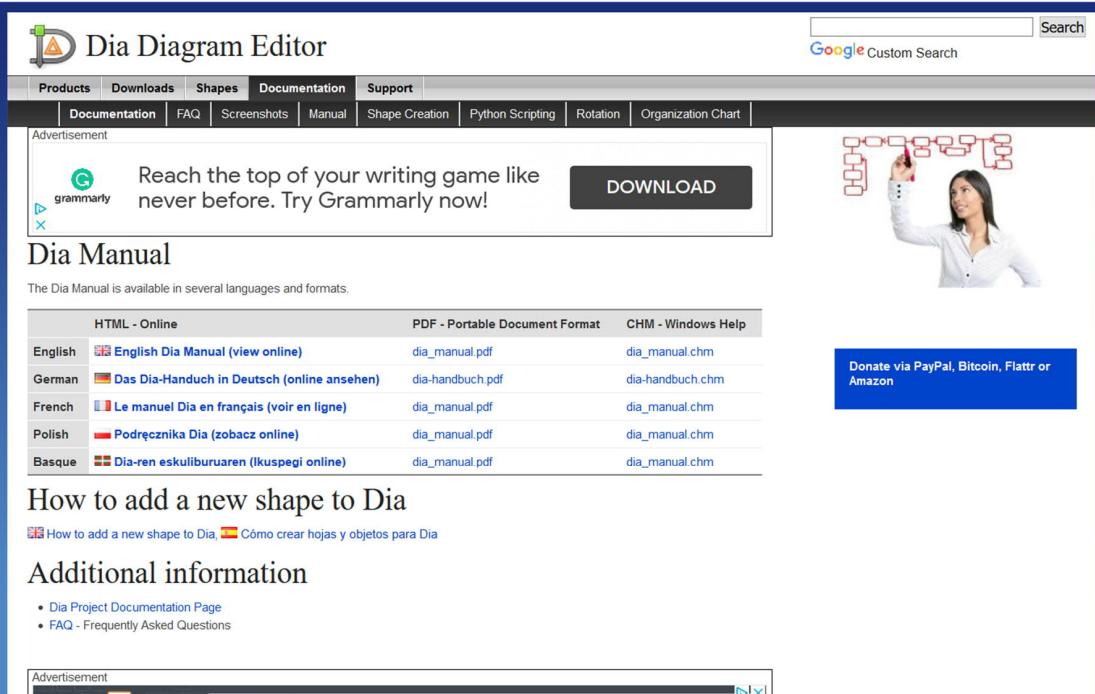
(a reprint of these slides is at Moodle; "Some Dia tool guidelines for ItSE 2020")

Dia diagramming tool

Dia is a simple free tool for creating UML diagrams. (Well, the best tools are never free.)

At TUNI software centre there are also two other diagramming tools, but **at EXAM classes Dia is installed** (also Enterprise Architect, more professional tool if you like to use it).

You are recommended to install Dia to your own computers before WE5 and WE6, and remember that at EXAM 2/3 you need to use Dia (or EA) tool.



The screenshot shows the Dia Diagram Editor documentation page. At the top, there's a navigation bar with links for Products, Downloads, Shapes, Documentation, and Support. Below the navigation bar, there's an advertisement for Grammarly and a section for the Dia Manual. The manual is available in several languages: English, German, French, Polish, and Basque. Each language has a link to the online manual, a PDF file, and a CHM file. To the right of the manual section, there's a blue button for donations via PayPal, Bitcoin, Flattr or Amazon. Below the manual section, there's a heading "How to add a new shape to Dia" with links to the English and Spanish versions. At the bottom of the page, there's an advertisement banner.

Some Dia hints

Dia is easy to use, and you can draw diagrams in many ways, the "right" way and many "wrong" ways. However if the result is understandable, it seldom can be known how the diagrams were actually drawn.

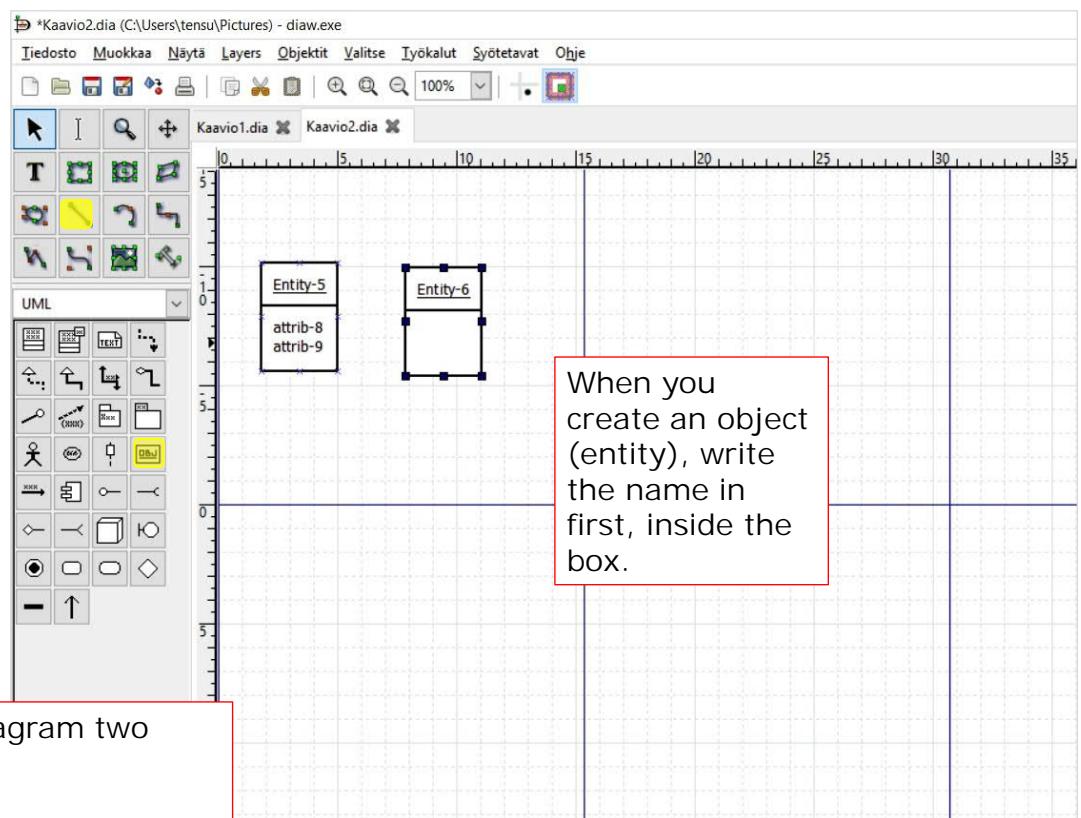
Using tools the right way, you may keep up with some method consistency checking embedded to the tool.

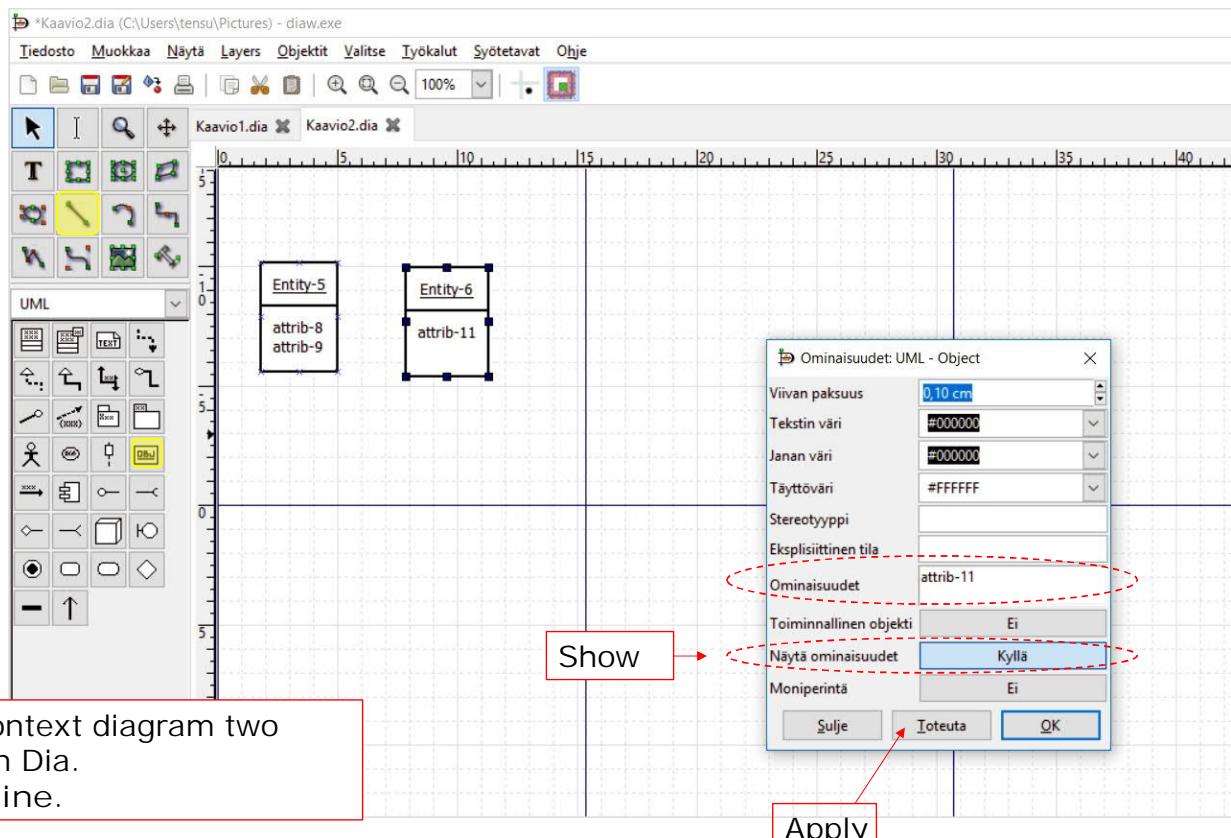
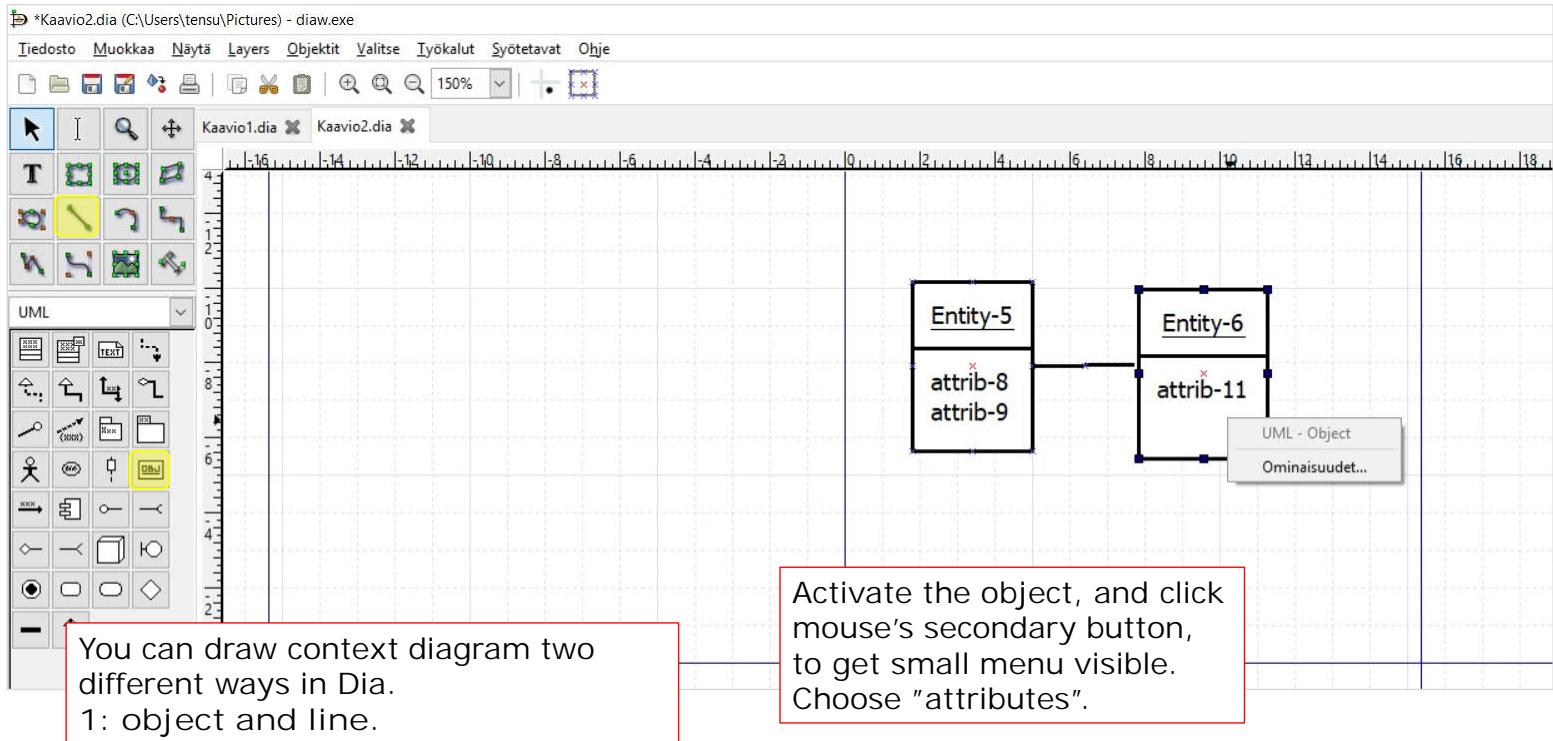
So here are a few simple hints for start of use.

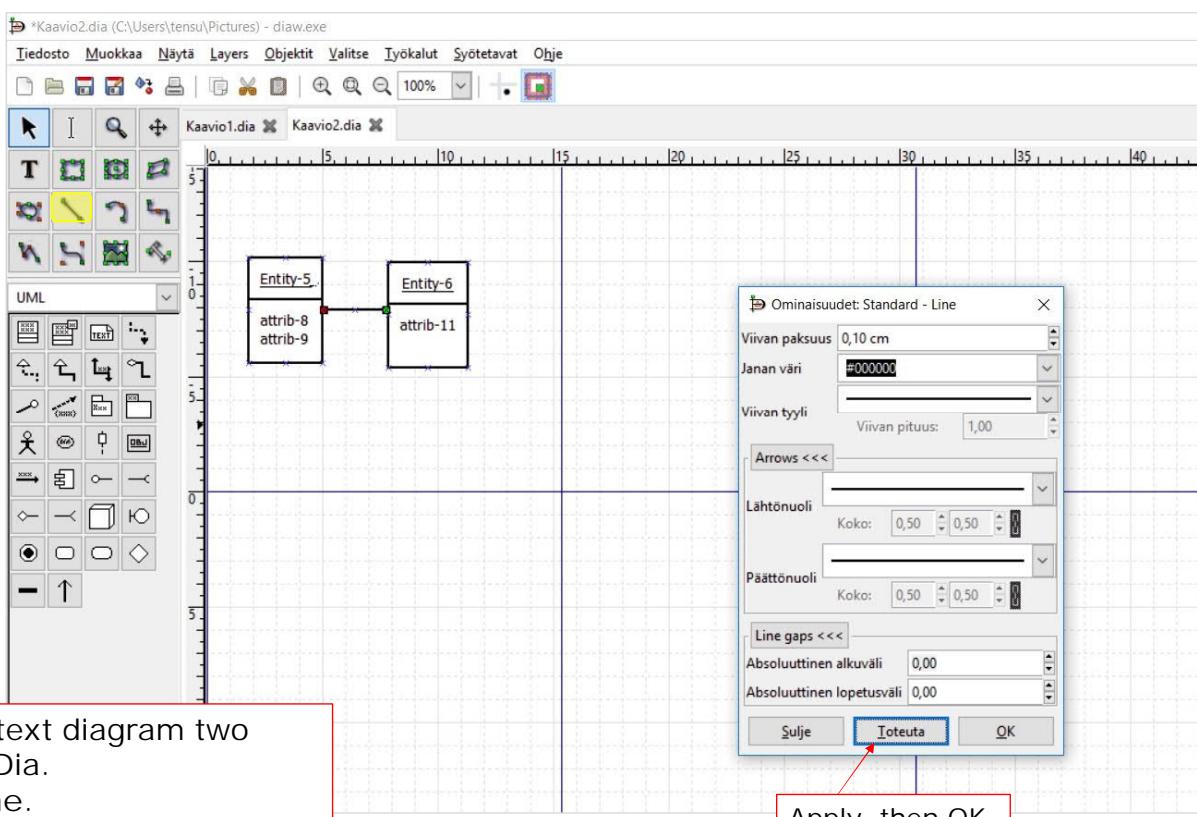
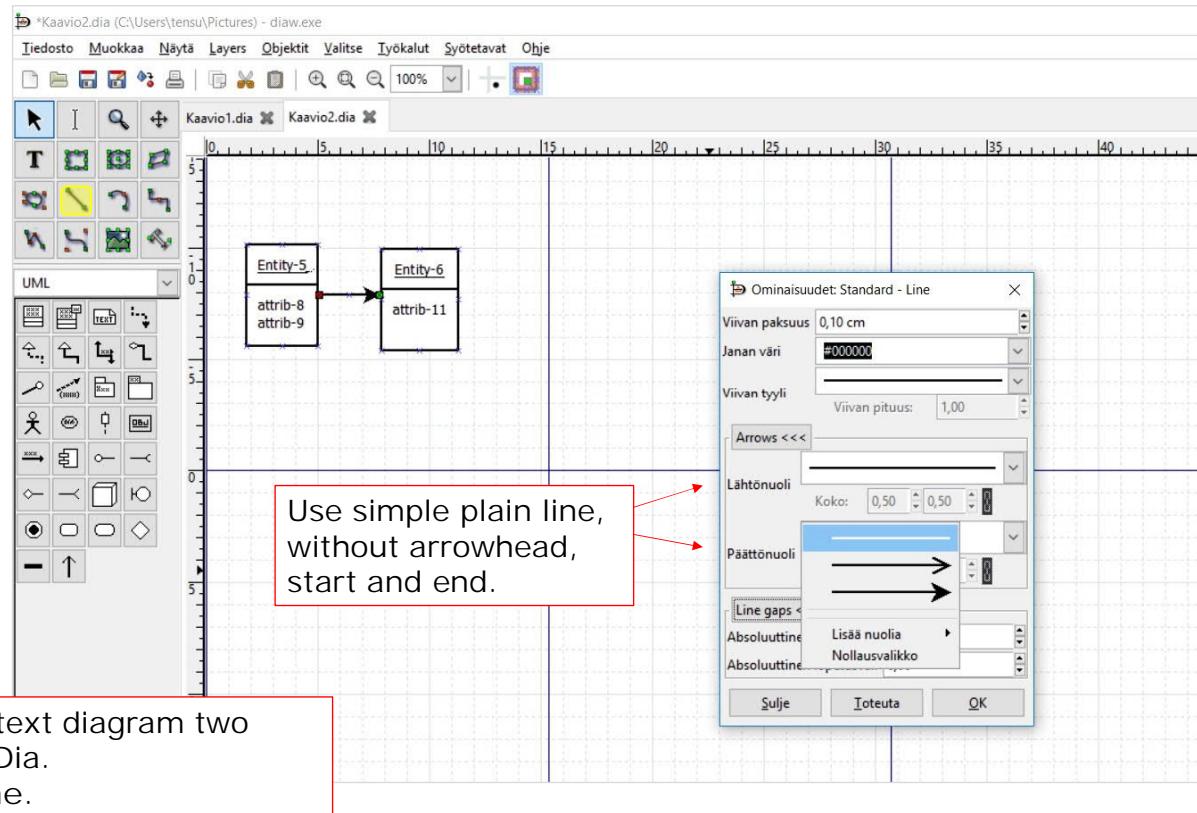
Sorry about my Dia version is in Finnish, that was done by some stupid automated installing. And I didn't had time to install another time (poor lazy lecturer).

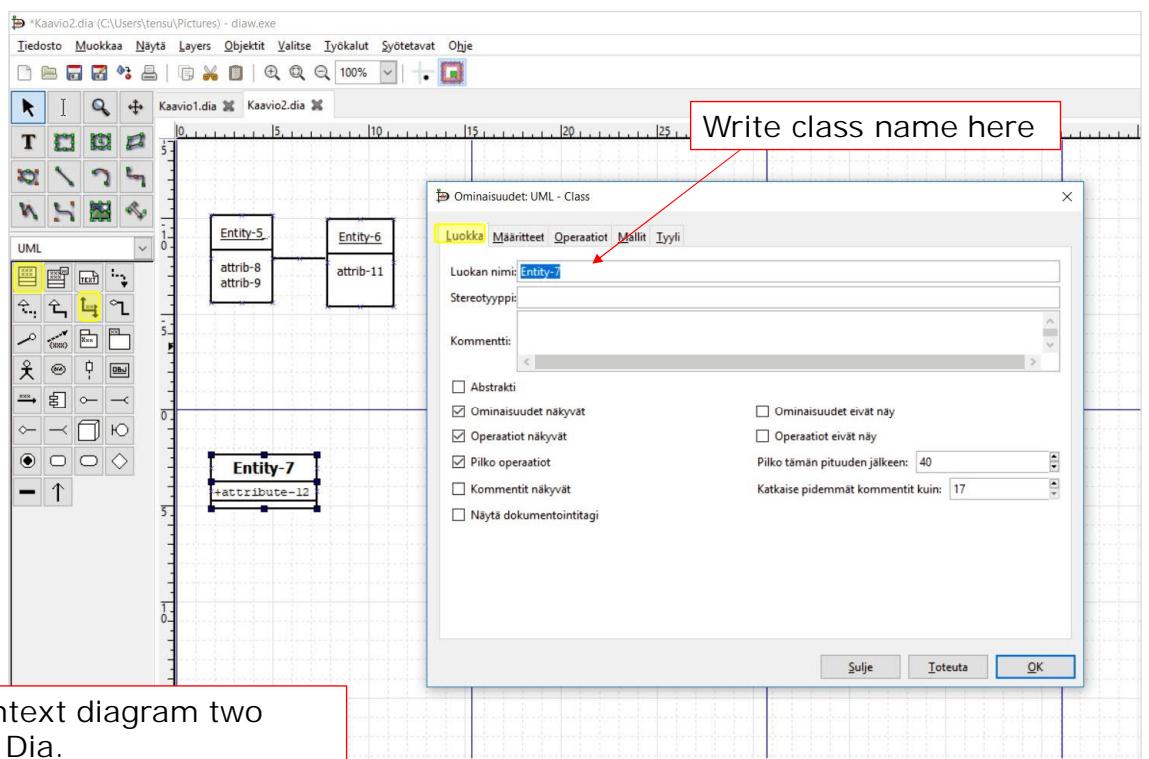
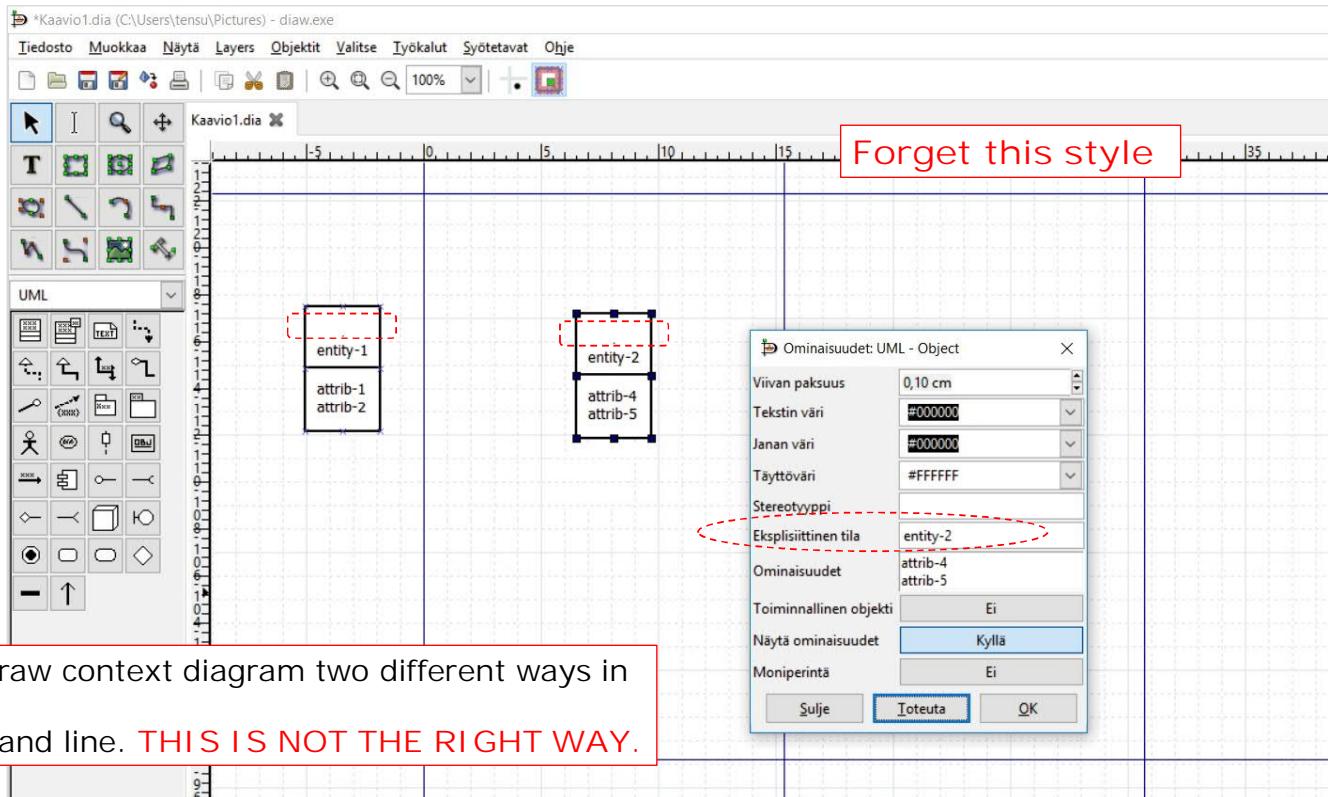
DIA, context/entity diagrams

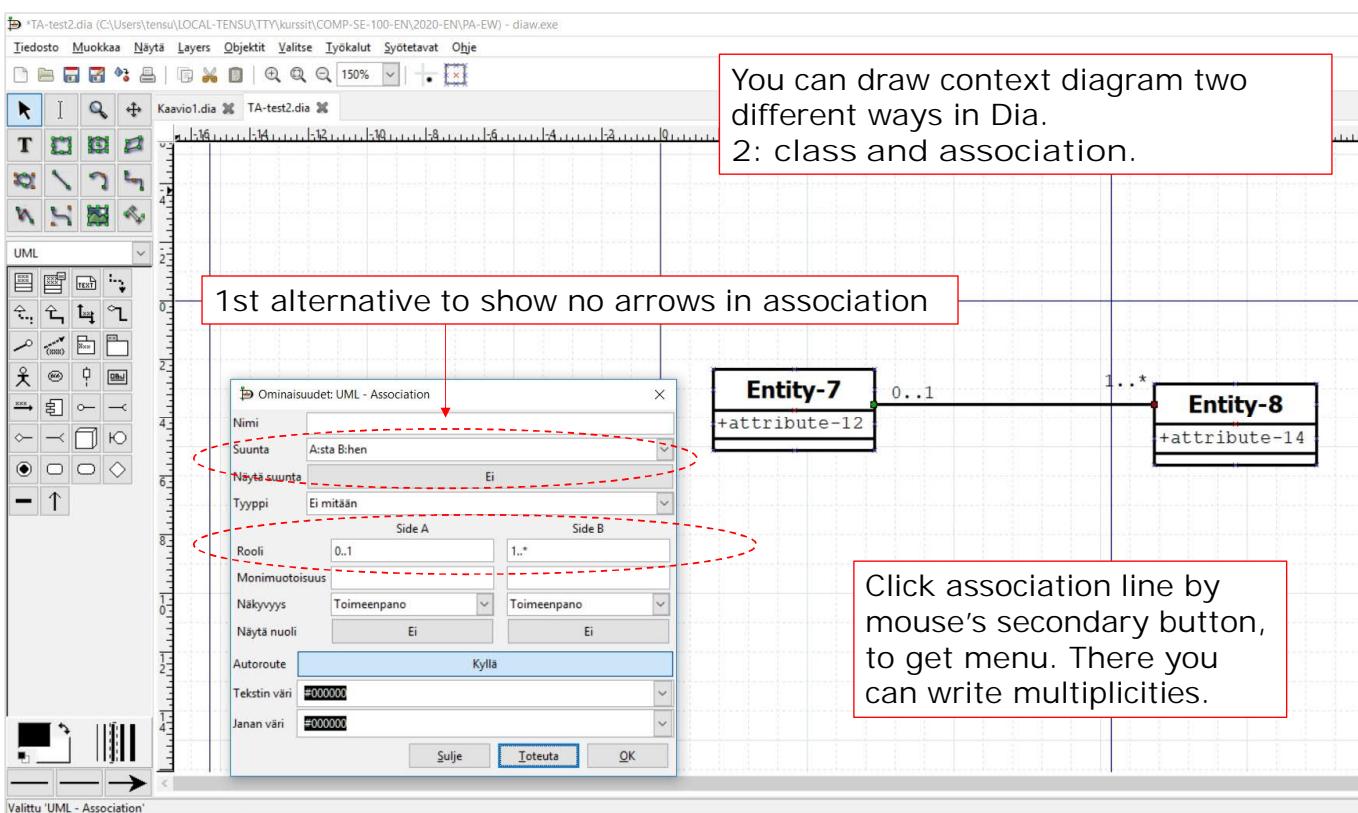
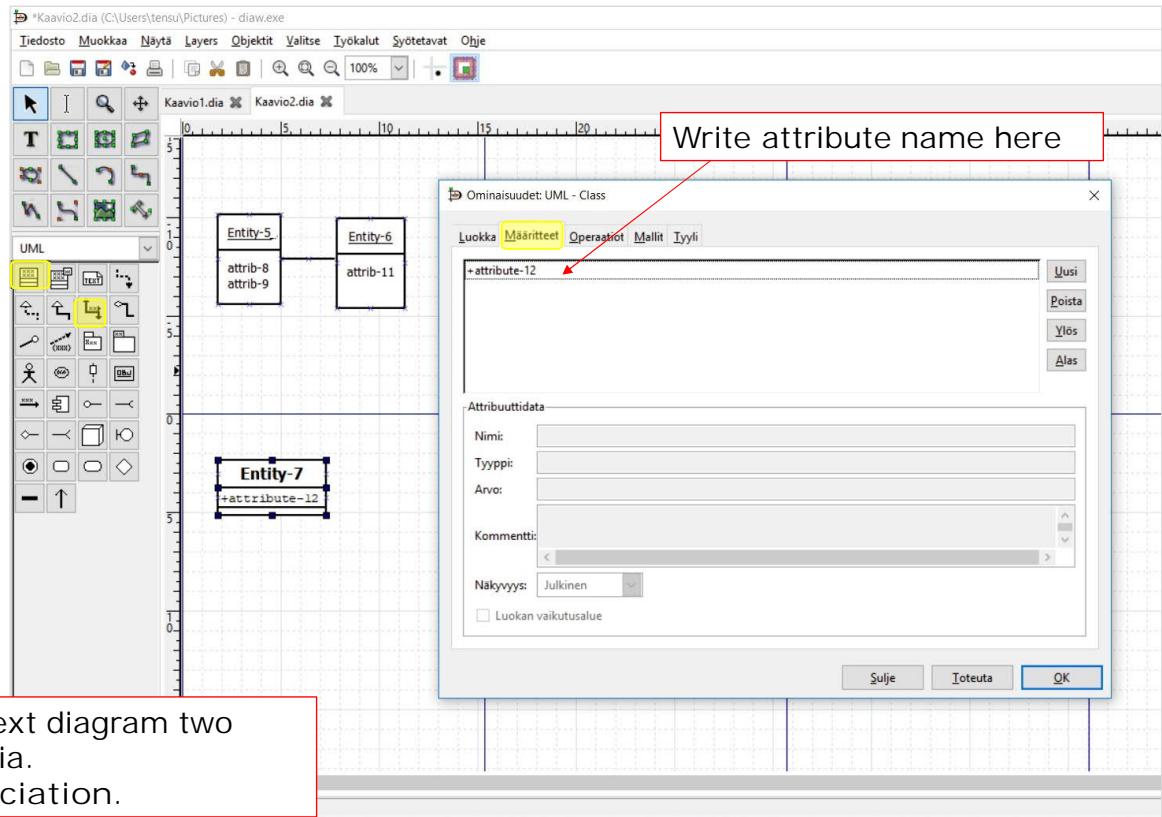
Use UML diagrams

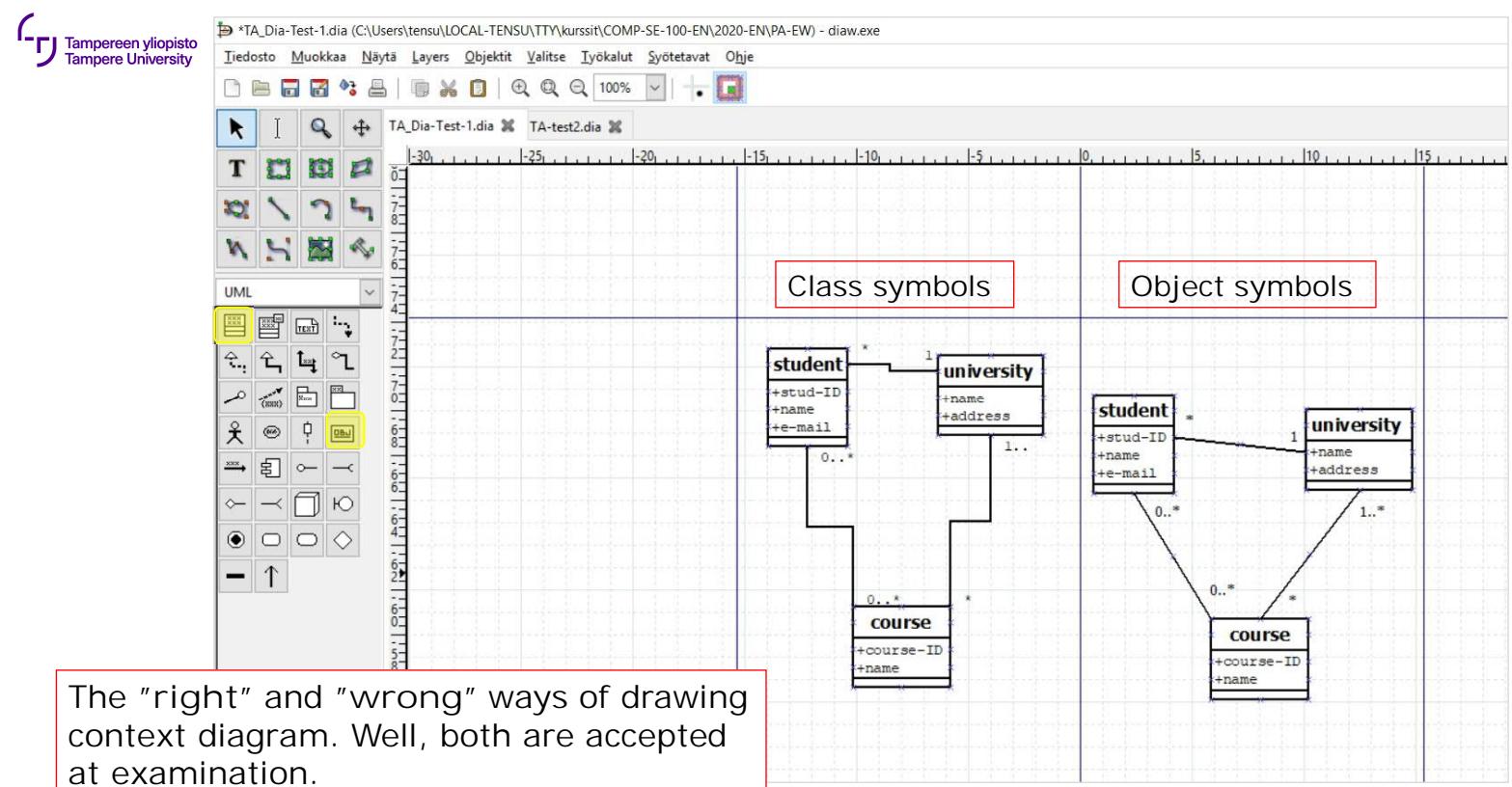
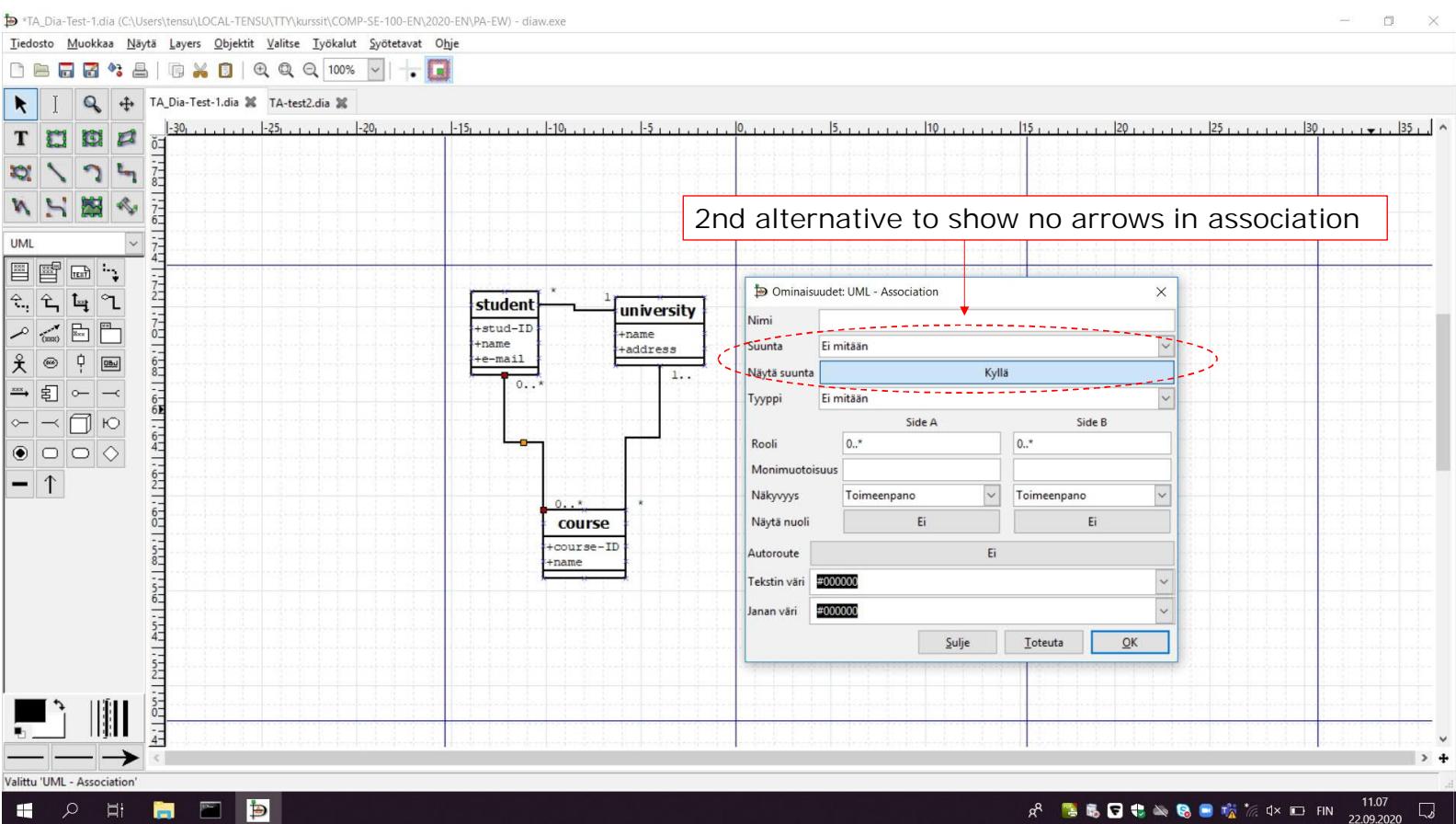


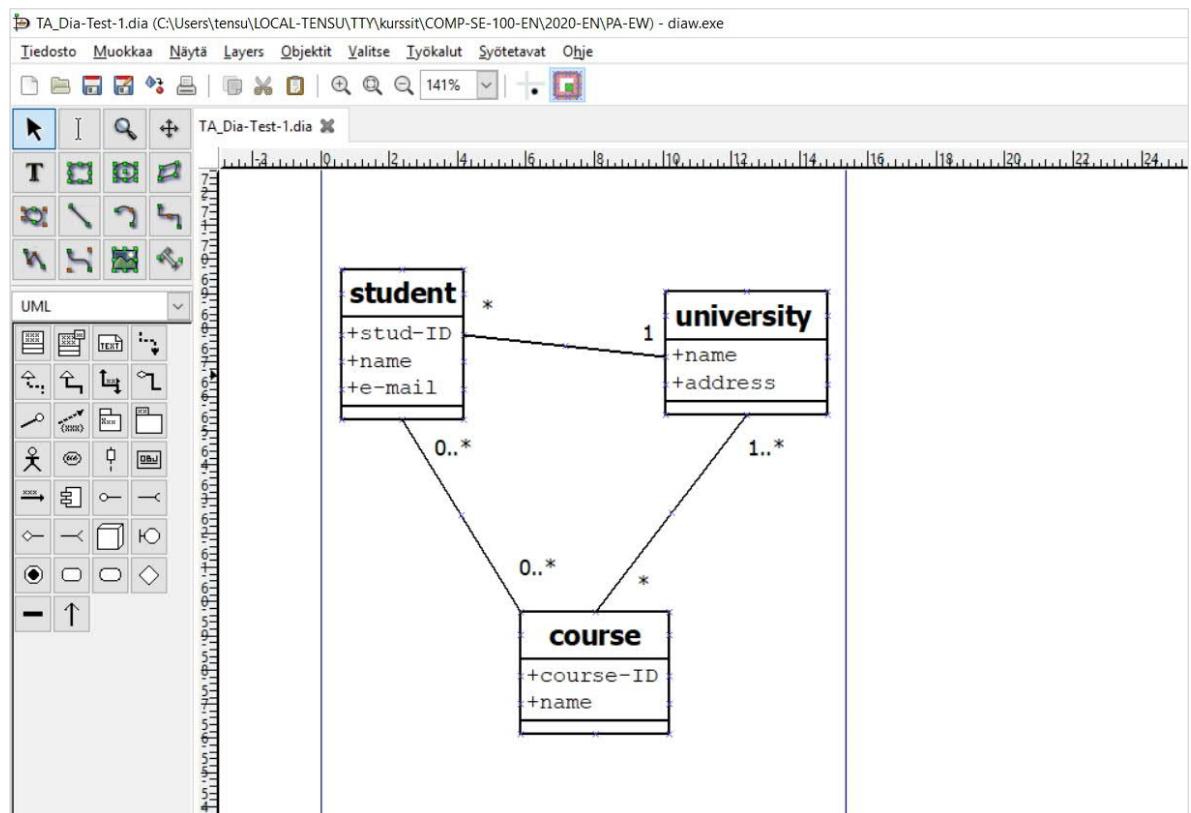
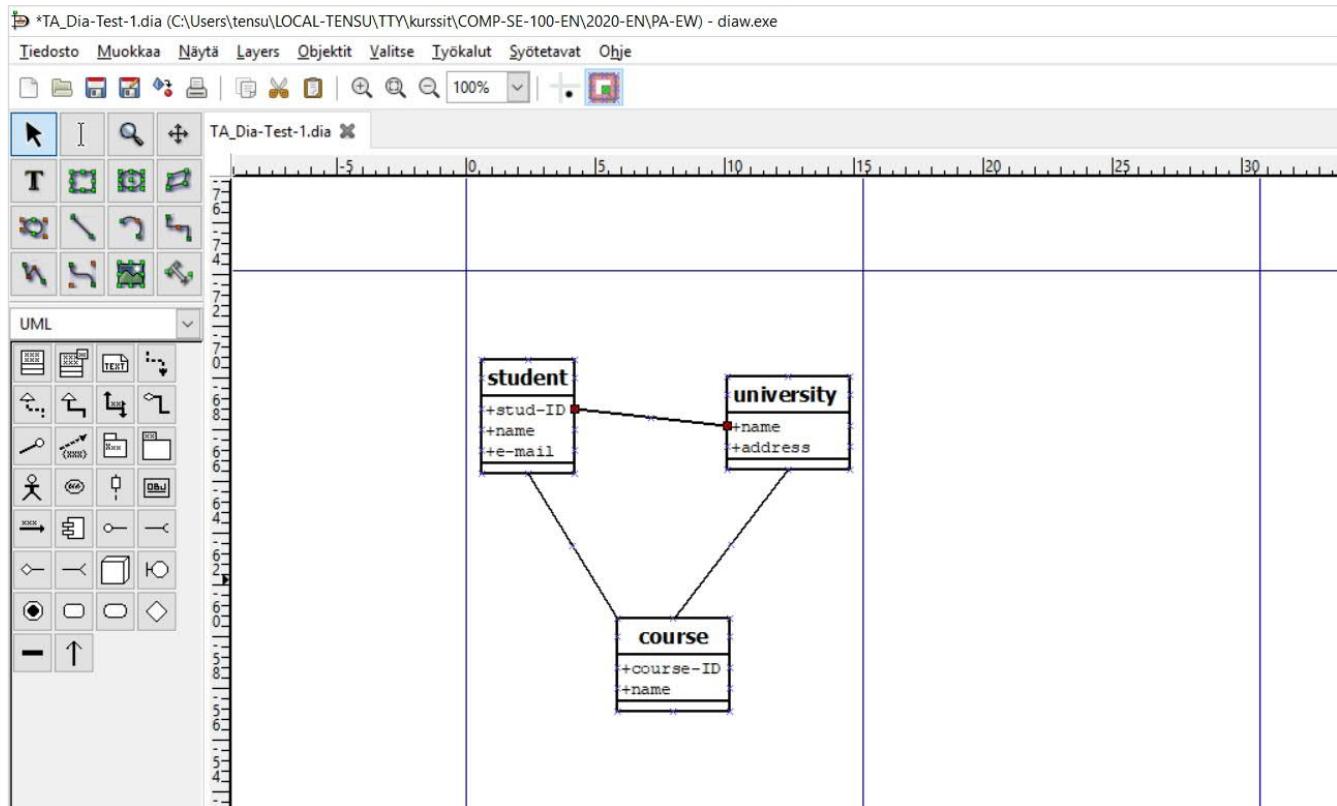


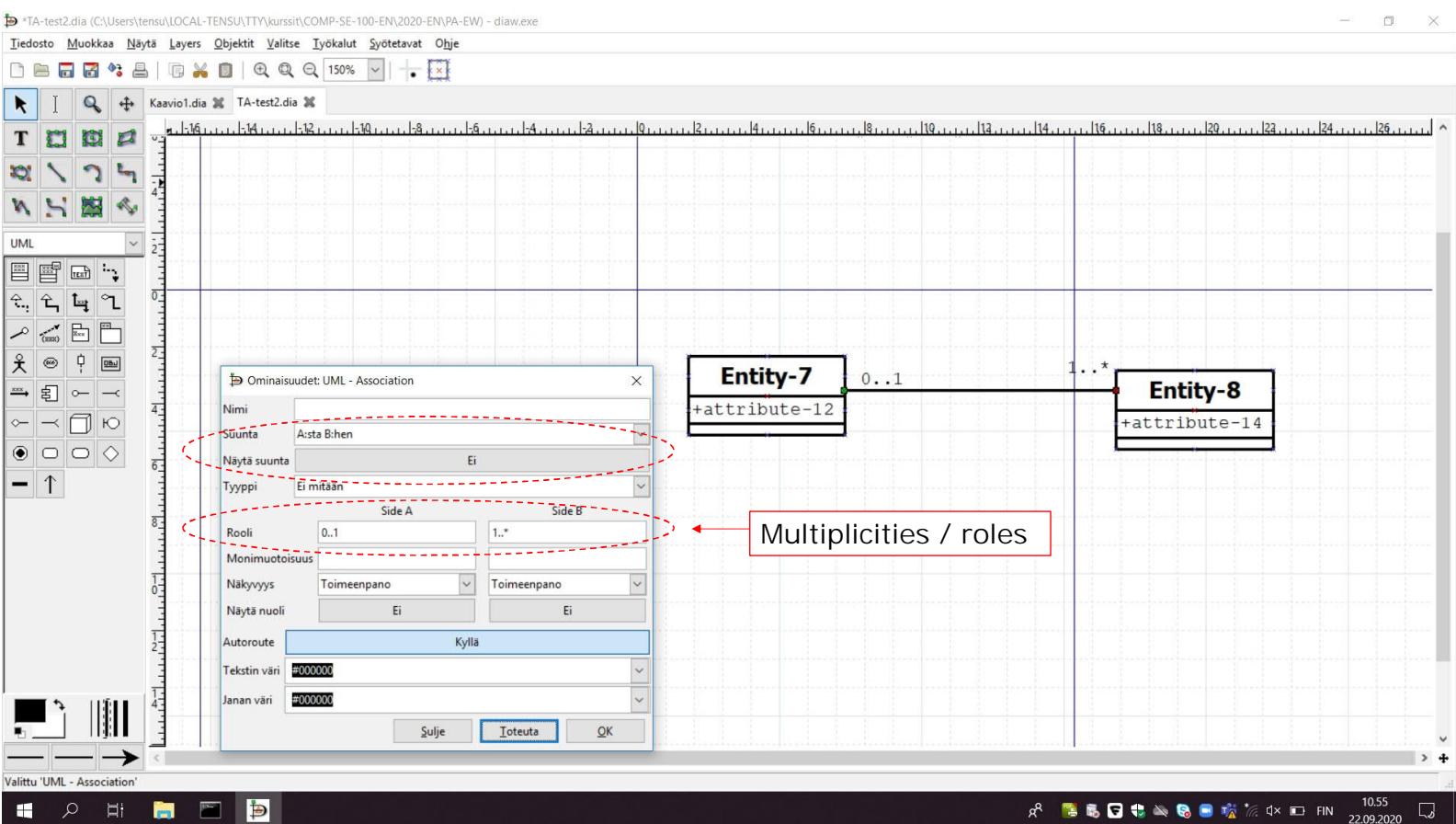




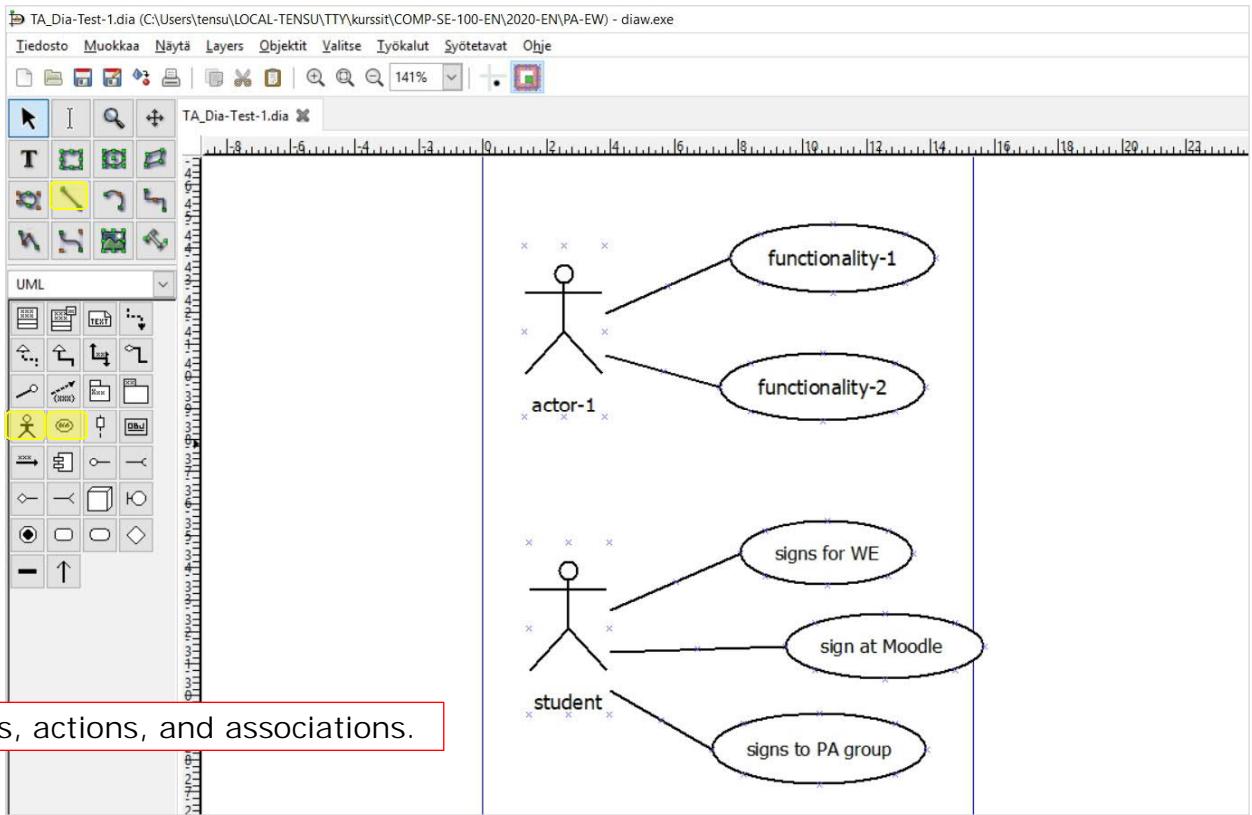




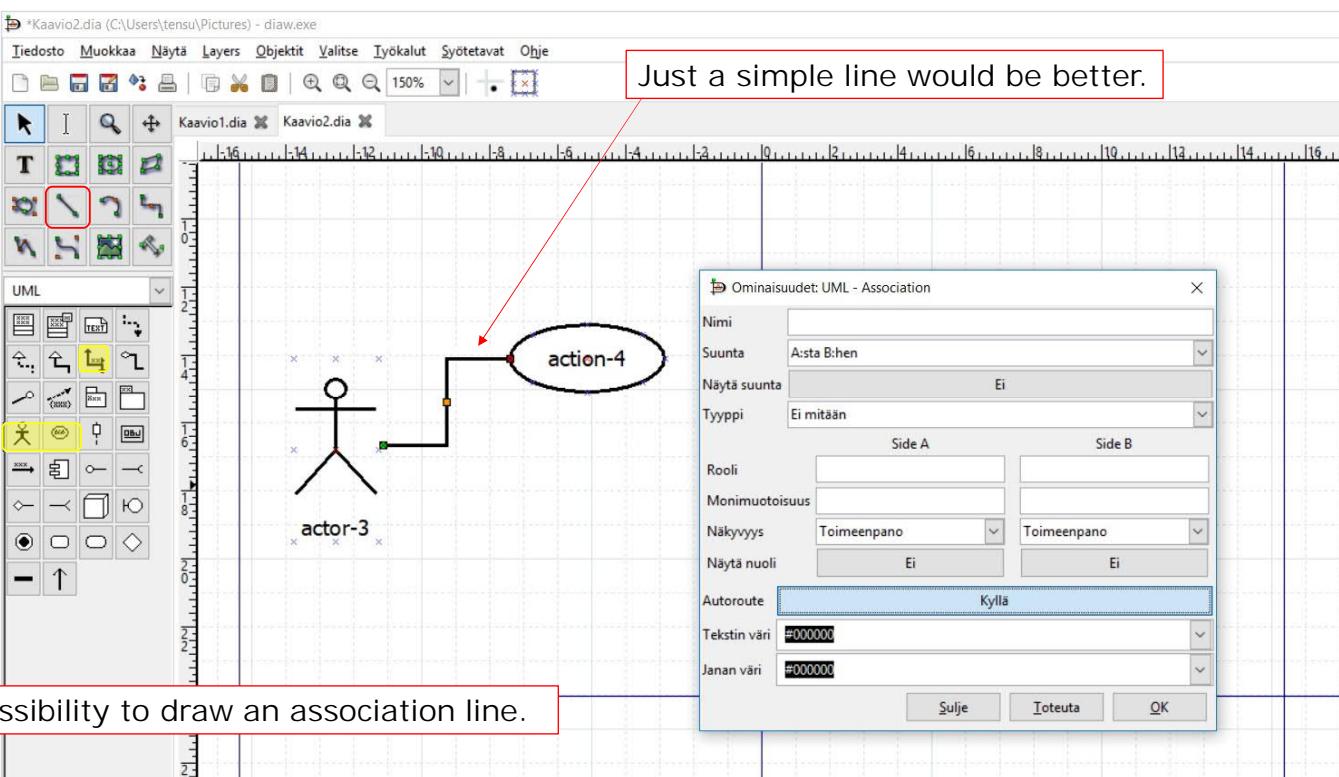




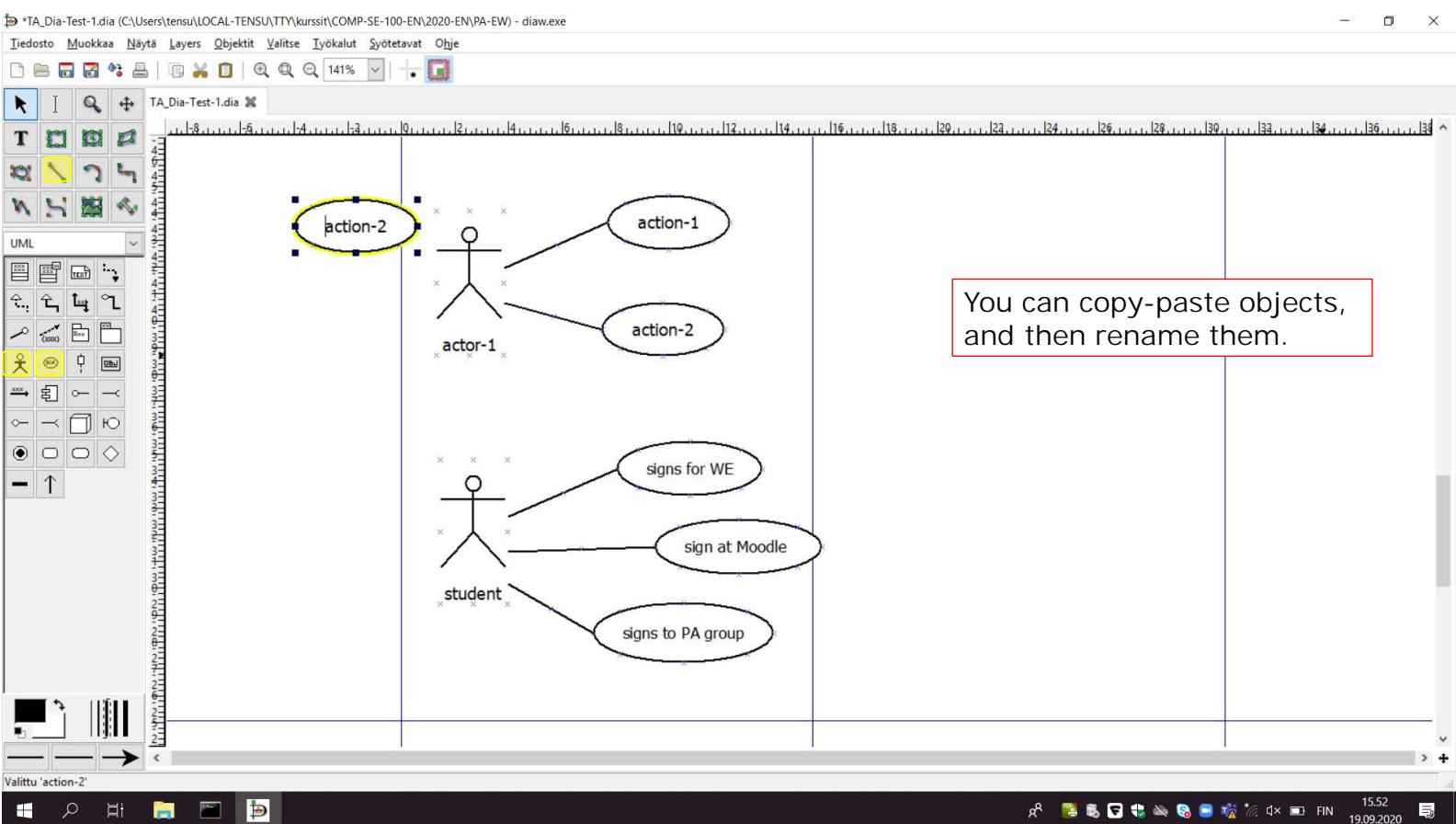
DIA, Use case diagrams



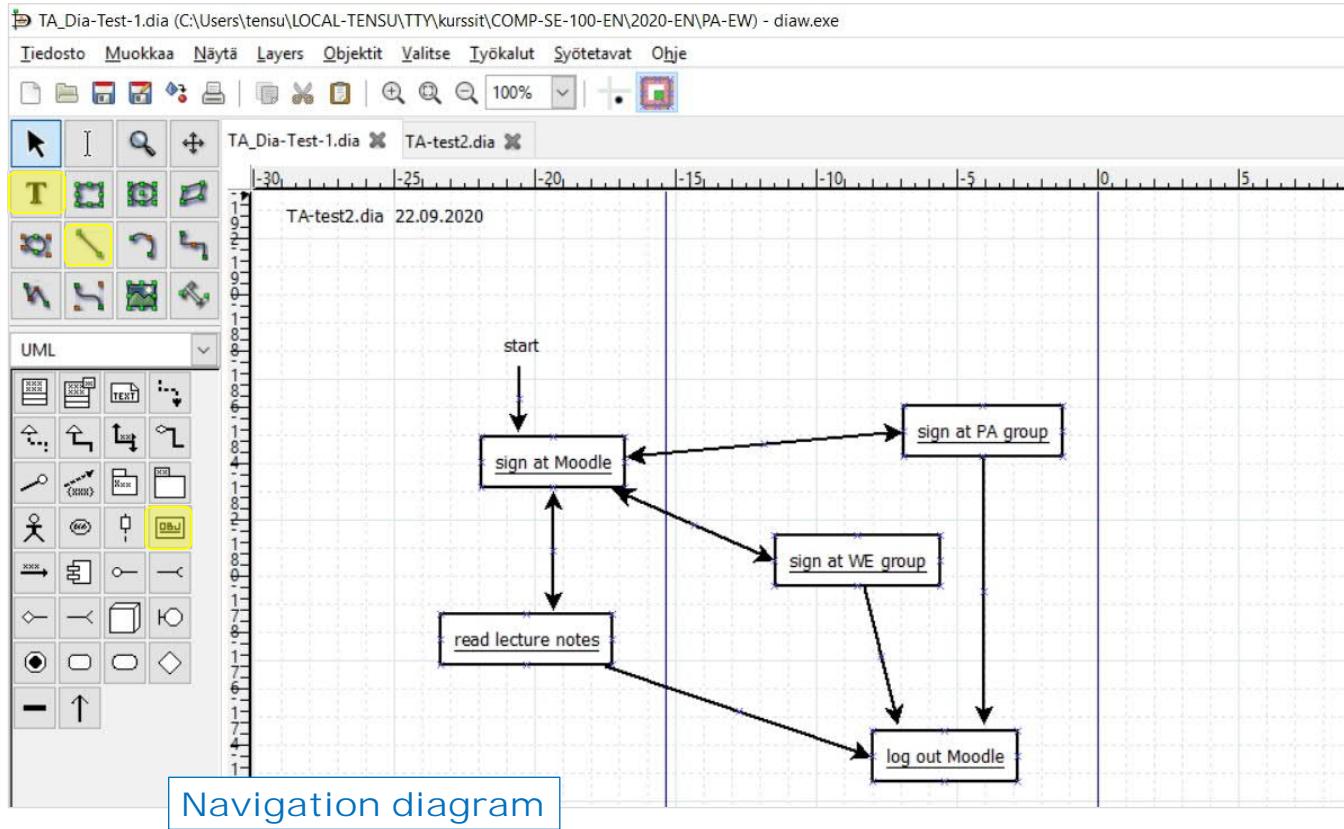
Draw actors, actions, and associations.



Another possibility to draw an association line.



Dia, navigation diagrams



COMP.SE.100-EN (ItSE) Introduction to Software Engineering

Lecture 3, 16.09.2020

Tensu: remember to pause
Zoom lecture recording

Zoom lecture break, 10 minutes stretching, walking, etc.

Basic UML Diagrams

- Entity diagram = context diagram, logical data model, "class diagram" (FI: **käsitekaavio, tietoyhteyskaavio, käsitemalli**), several versions exist depending of amount of information content **terminology is not coherent**
- Use Case diagram (FI: **käyttötapauskaavio**) and User Stories (FI: **käyttäjatarinat**)
- Navigation diagram (FI: **navigointikaavio, navigatiokaavio**) (not officially UML).

Those three diagram types you need at project assignment (exercise work), and at EXAM 2/3.

Entity / context diagram

Entity / context (conceptual) diagram

In many cases, also in Finnish, this diagram type is called "Class Diagram", just because it is drawn by "class diagram" symbols and functionality in tools.

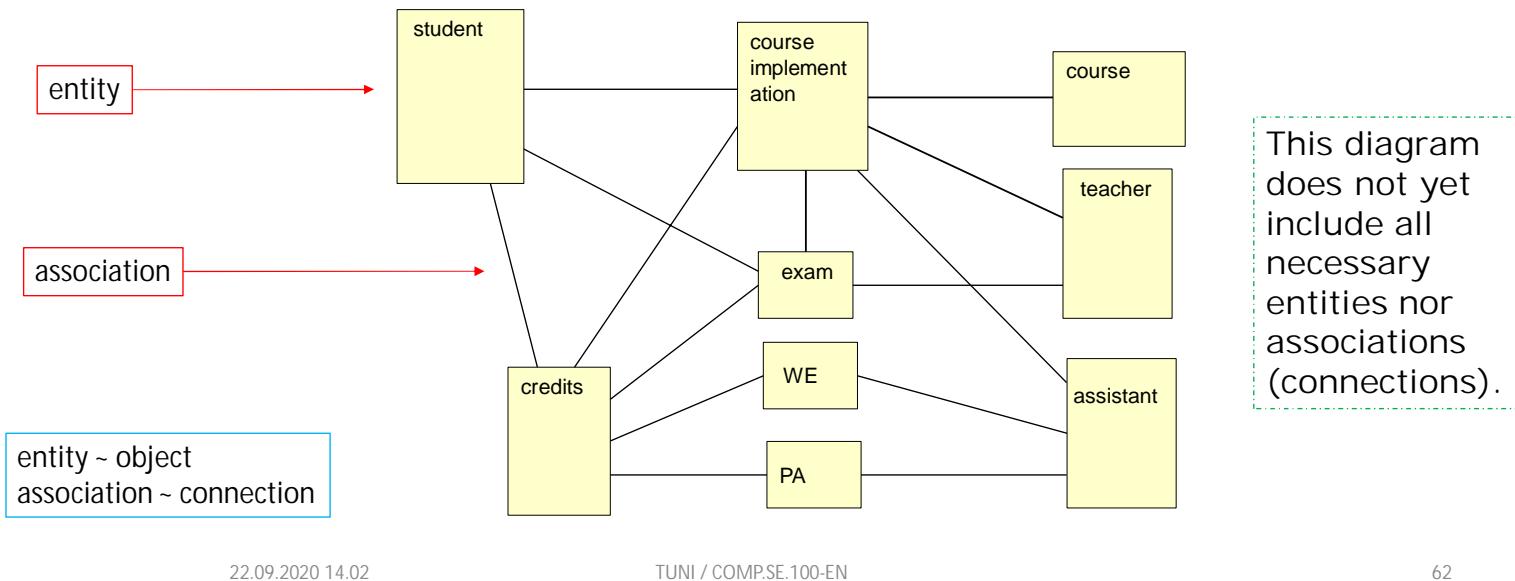
This diagram type is the very first used when starting a project. By entity (or context) diagram you get an **overall idea about the application area** and environment. It is something like a mind-map.

The idea is to first draw (write) all entities you know, and then connect together those which are related or affect each other. After that, you think carefully which entities are actually needed at the system, and which you may drop out.

By the way, at this point it makes sense to make a (data) dictionary, to avoid misunderstandings. For example, at university, what "class" or "course" means ?

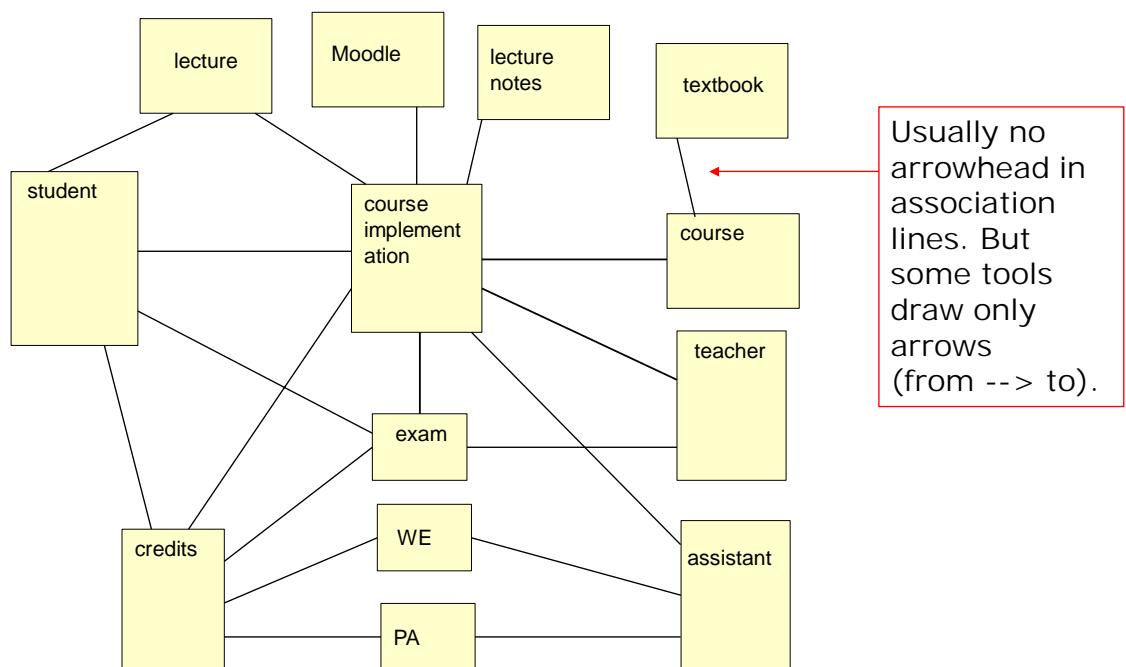
(Actually class diagram is used in design phase, helping to define which attributes and operations/methods an object should have in code.)

Entity/Context diagram, about University course. What is still missing ?



Entity/Context diagram.

This diagram does not even yet include all necessary entities nor associations (connections) to be complete.



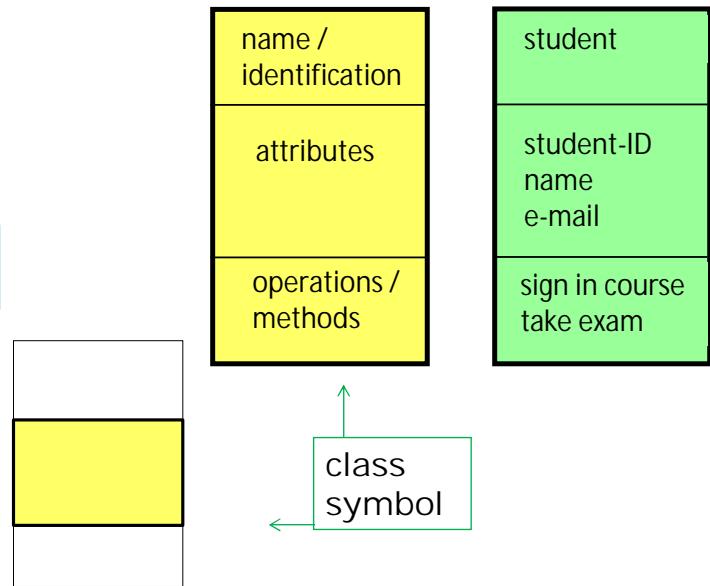
Entity/Context (class) diagram

Terminology is not coherent, may vary depending on the speaker.

- entity, instance, object, class,...

Diagramming tools draw classes as three-part boxes.
Do not care about the third (lower) part.

attributes
(middle part)

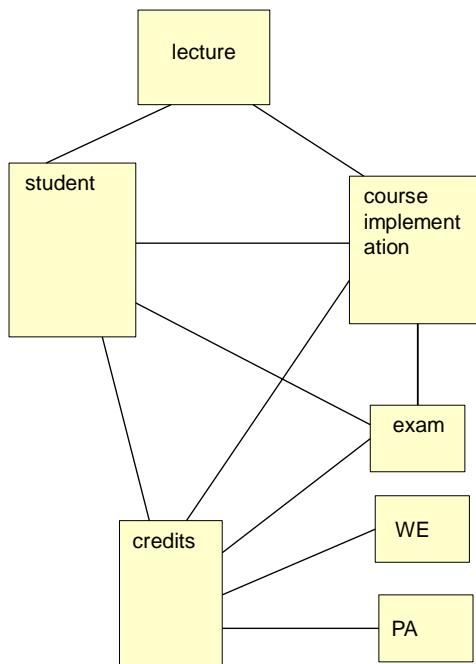


22.09.2020 14.02

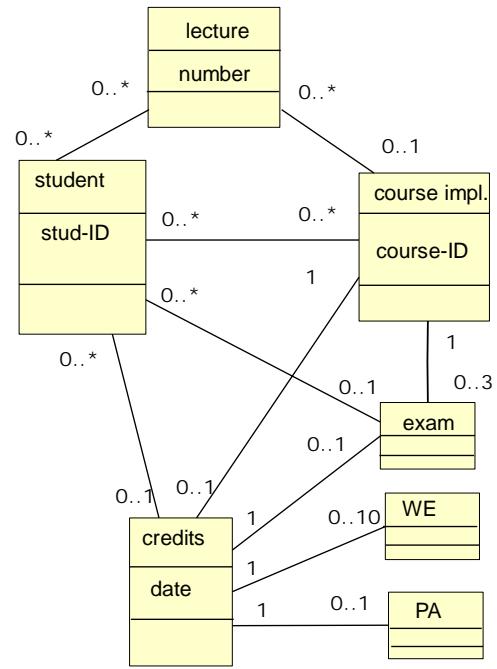
TUNI / COMP.SE.100-EN

64

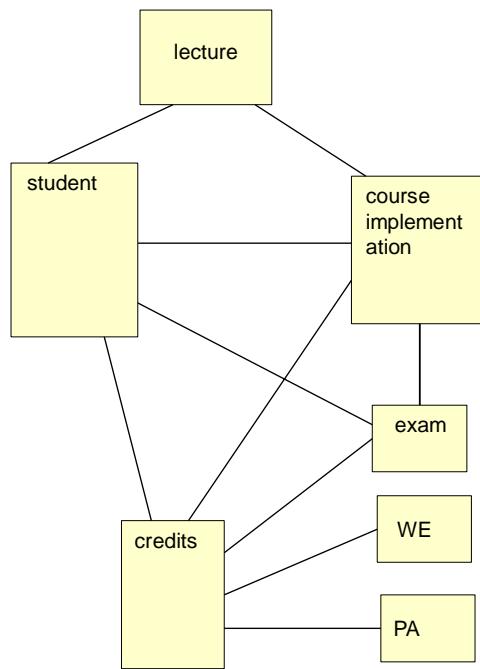
Entity diagram (FI: käsitekaavio)



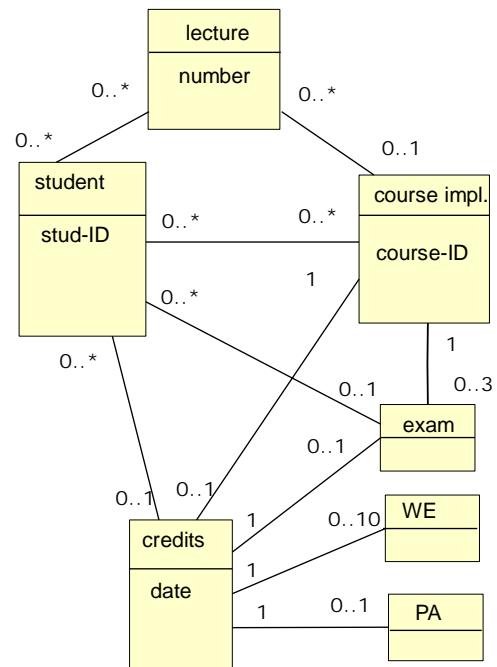
Class diagram (FI: luokkakaavio)



Entity diagram (FI: käsitekaavio)



(FI: tietoyhteyskaavio)



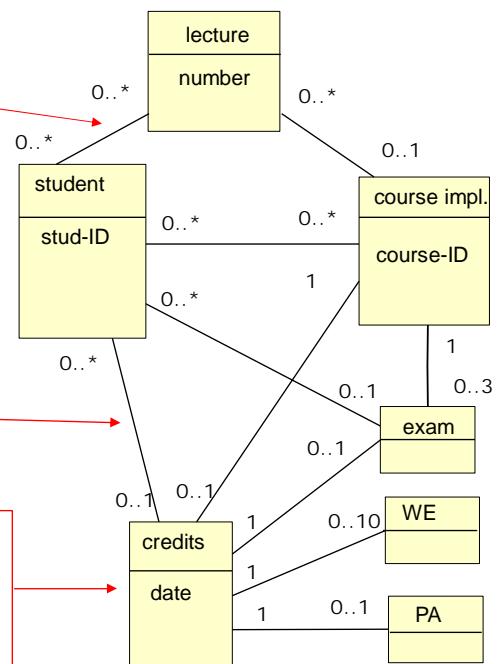
22.09.2020 14.02

TUNI / COMP.SE.100-EN

66

Multiplicities (FI: lukumääräsuhteet)

Student takes part or not to one or more lectures = $0..*$ $0..*$
At lectures there are zero or more students.



Zero or more students has credits,
Student has one or has not (zero)
credits.

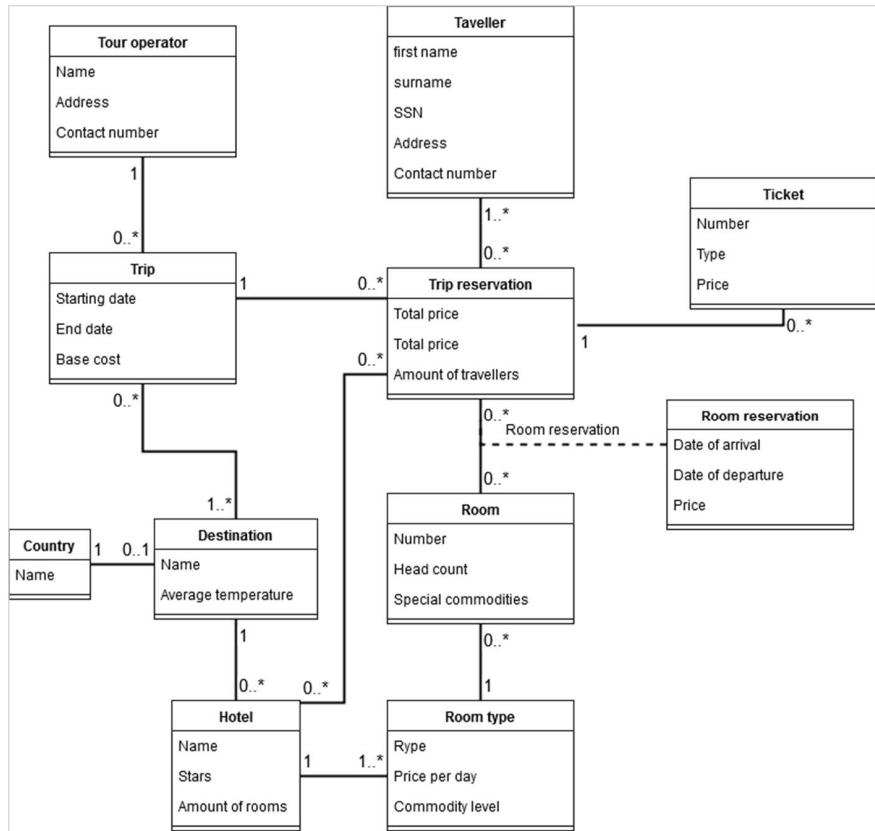
To one (student's) credits belong
zero or no exam,
zero to ten weekly exercises
zero or one project assignment.
All exam, WE, PA info is connected to some credits.
Credits belong to one course implementation.

22.09.2020 14.02

TUNI / COMP.SE.100-EN

67

Travel agency

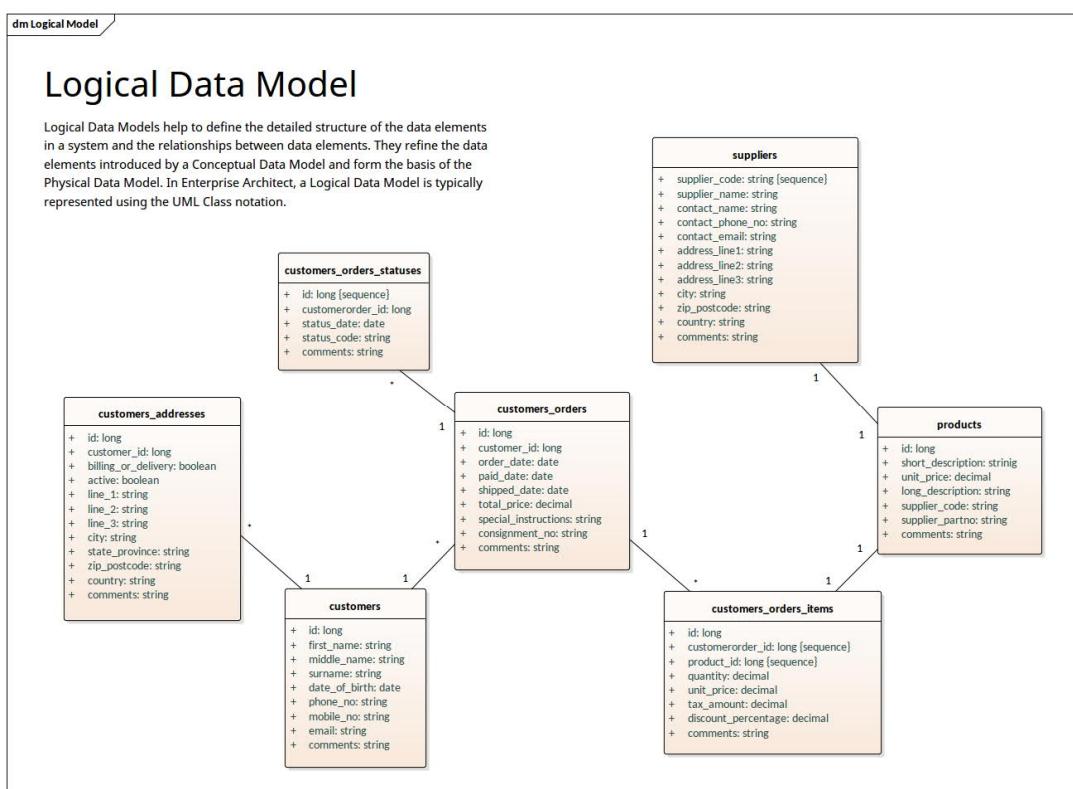


TUNI / COMP.SE.100-EN

22.09.2020 14.02

68

"+" sign in front of attributes is from the tool



TUNI / COMP SE 100-EN

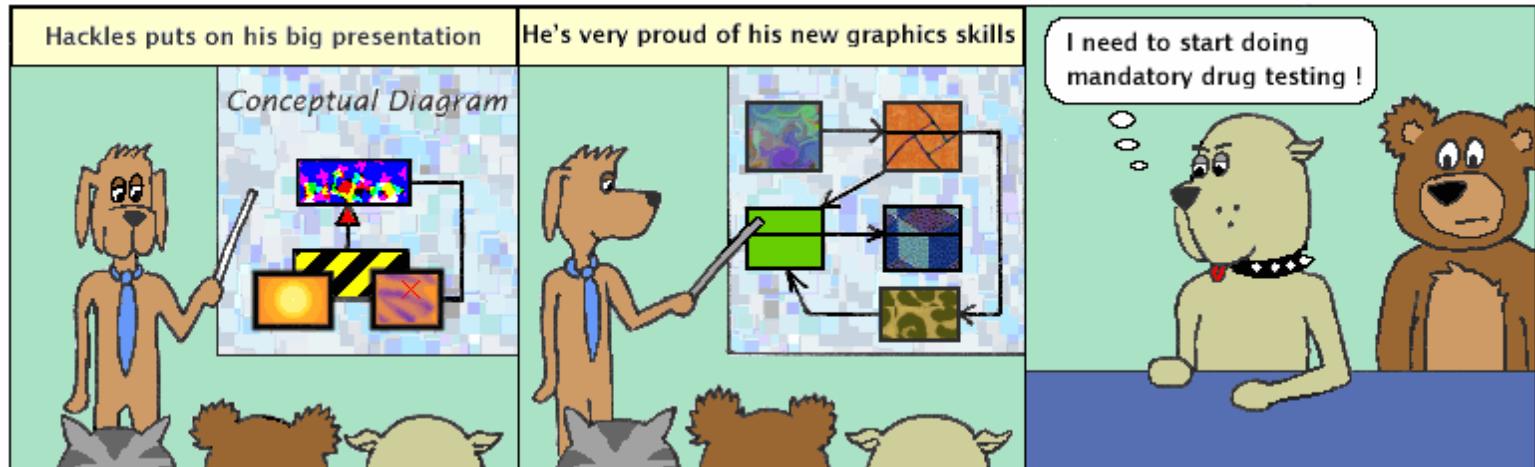
[sparxsystems.com/resources/gallery/diagrams/]

22.09.2020 14:03

69

Hackles

By Drake Emko & Jen Brodzik

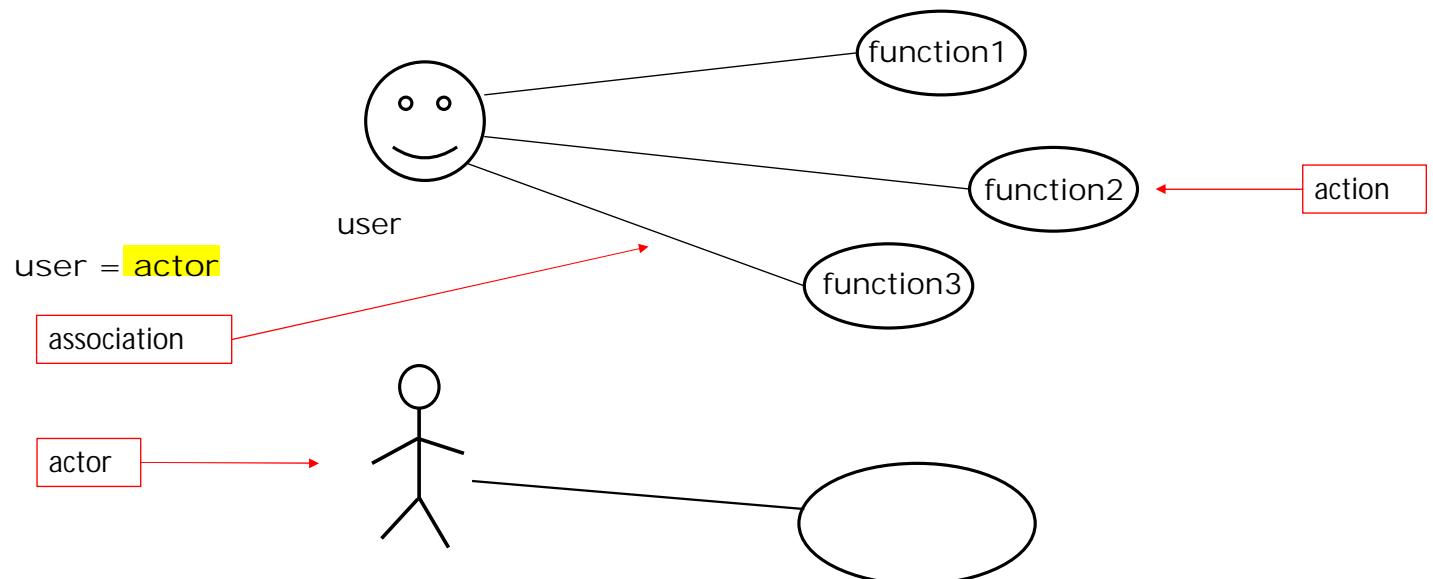


<http://huckles.org>

Copyright © 2001 Drake Emko & Jen Brodzik

Use case diagram

Use Case diagram is usually made at the first phases of project, to help at requirements gathering

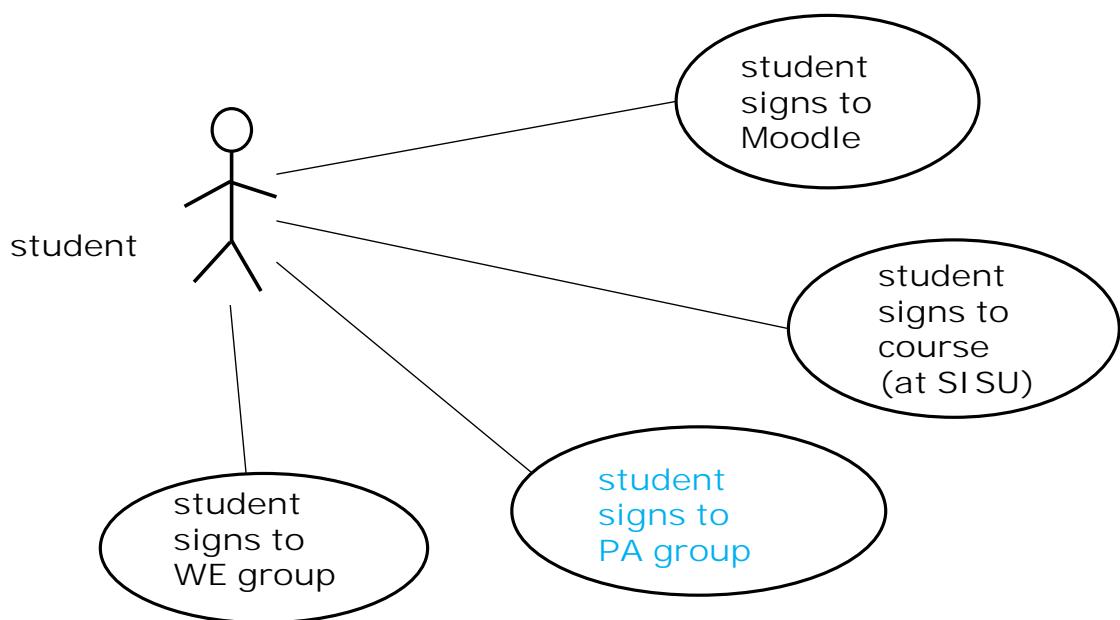


22.09.2020 14.02

TUNI / COMP.SE.100-EN

72

Use Case diagram, university course example



22.09.2020 14.02

TUNI / COMP.SE.100-EN

73

User Story (Use Case), one function at Use Case diagram

User story for "Student signs to PA group", student signs for project assignment (exercise work):

- Actor: student.
- Start state: Student is signed to course at SISU. Student is signed at Moodle. Student has selected course implementation (e.g. ItSE 2020).
- Actions: Student selects signing to PA. Student looks at groups and selects one. Student signs to that group. Student gets confirmation message.
- End state: student is signed at one PA group.
- Exceptions: No "free seats" at any PA group, all groups are full. Session timeout.
- Priority: required.

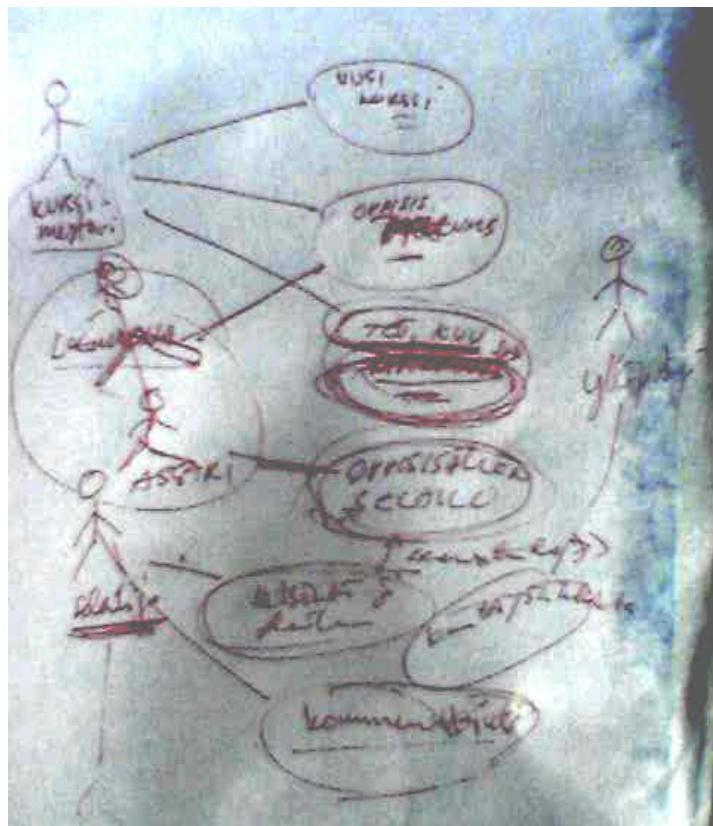
22.09.2020 14.02

TUNI / COMP.SE.100-EN

74

Quick Use Case sketching to a napkin at restaurant when meeting with vendor and customer.

(is this perhaps POP/ROCK or SISU...)



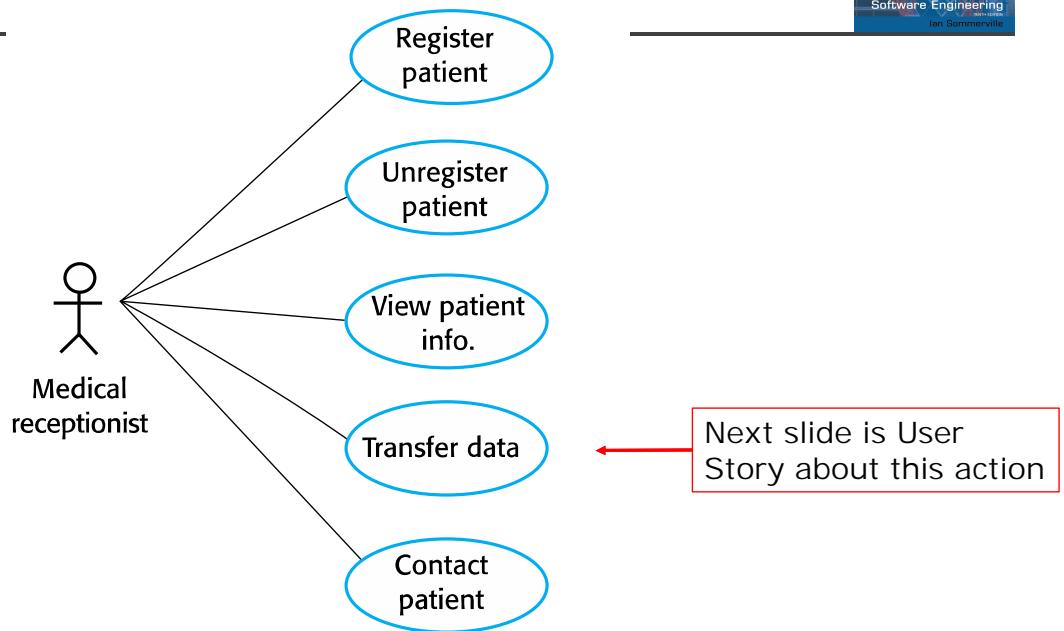
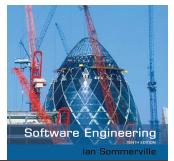
22.09.2020 14.02

TAU / TIE-02301

75



Use cases in the Mentcare system involving the role 'Medical Receptionist'



22.09.2020 14.02

TUNI / COMP.SE.100-EN

76

Tabular description of the 'Transfer data' use-case



MHC-PMS: Transfer data	
Actors	Medical receptionist, patient records system (PRS)
Description	A receptionist may transfer data from the Mentcase system to a general patient record database that is maintained by a health authority. The information transferred may either be updated personal information (address, phone number, etc.) or a summary of the patient's diagnosis and treatment.
Data	Patient's personal information, treatment summary
Stimulus	User command issued by medical receptionist
Response	Confirmation that PRS has been updated
Comments	The receptionist must have appropriate security permissions to access the patient information and the PRS.

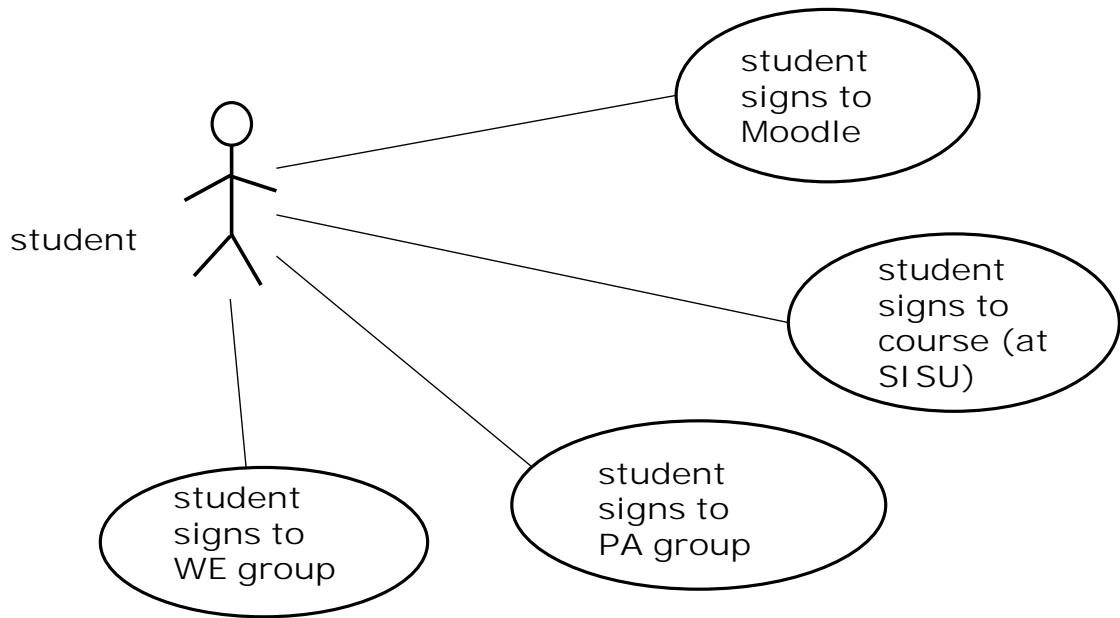
22.09.2020 14.02

TUNI / COMP.SE.100-EN

77

Use Case diagram, you can not see from that

- dependencies between entities
- what data flows or is in the system
- chronology (order) for the functions.

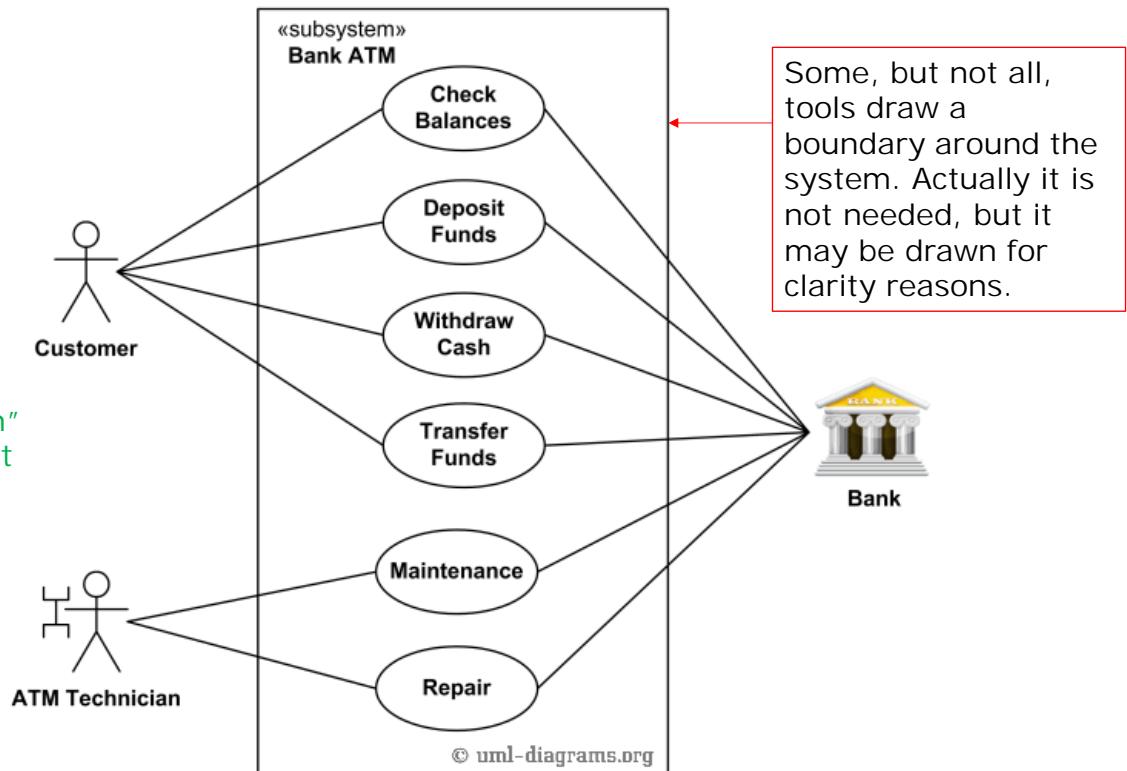


22.09.2020 14.02

TUNI / COMP.SE.100-EN

78

An example of use case diagram for Bank ATM subsystem - top level use cases.



This may look "childish" simple, but it tells what are the users, and which user groups are doing what.

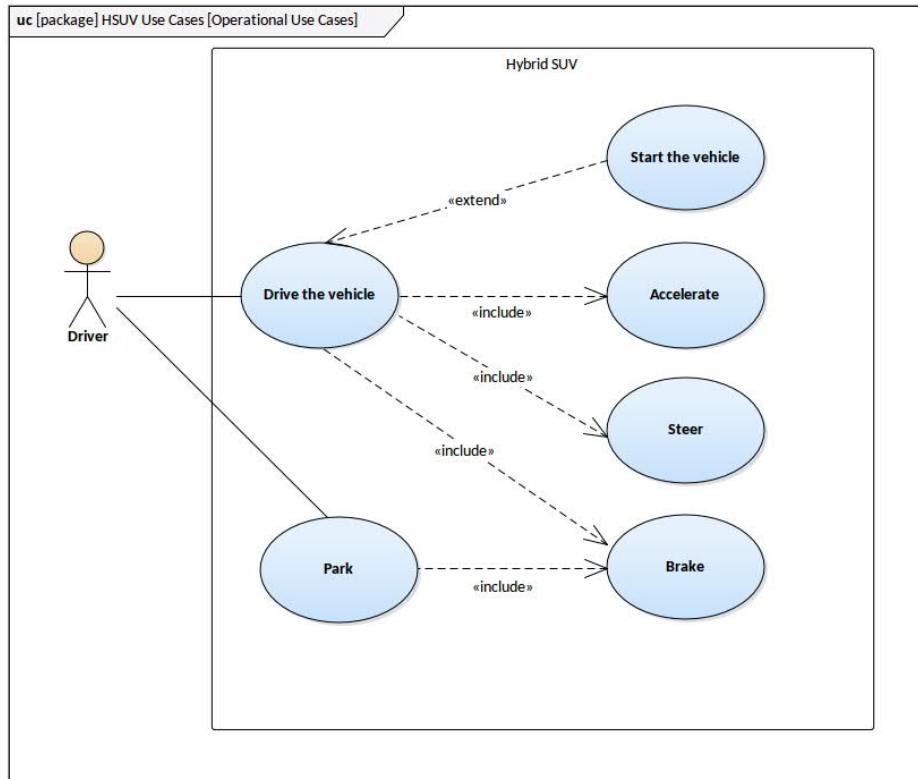
22.09.2020 14.02

TUNI / COMP.SE.100-EN

79

Use Case diagram,

example of **include** and **extend**, but do not use these if you are not sure what you do.



22.09.2020 14.02

[sparxsystems.com/resources/gallery/diagrams/]
TUNI / COMP.SE.100-EN

80

User stories (Mike Cohn)

A user story describes functionality that will be valuable to either user or purchaser of software.

User stories are composed of three aspects:

- A written description of the story, used for planning and reminder
- Conversations about the story to serve to flesh the details of the story
- Tests that convey and document details and that will be used to determine that the story is complete.

What are characteristics of a good user story ?

The acronym "INVEST" can remind you that good stories are:

I – Independent

N – Negotiable

V – Valuable

E – Estimable

S – Small

T – Testable.

another INVEST

There are some helpful frameworks to help for writing strong user stories. One of the most well-known of these frameworks is a mnemonic called INVEST, created by consultant and developer Bill Wake:

Independent: It should be self-contained (i.e., not dependent on another user story).

Negotiable: There should be room for discussion.

Valuable: The story must provide value to stakeholders.

Estimable: The amount of effort to implement the story's functionality can be estimated.

Small: It should be completable in a single sprint.

Testable: The story must include enough detail to allow the creation of tests that verify the functionality the narrative addresses.

A user story template is a common format used to write user stories that helps you include key pieces of information about that user story.

One particular template, often referred to as “**As a... I want to... So That...**” is the most commonly recommended aids (often outgrown once past the novice stage) for teams and product owners starting to work with user stories and product backlog items in general:

- As a (who wants to accomplish something)
- I want to (what they want to accomplish)
- So that (why they want to accomplish that thing).

An example:

- As a bank customer
- I want to withdraw money from an ATM
- So that I’m not constrained by opening hours or lines at the teller’s.

AGILE USER STORY TEMPLATE

USER STORY ID	As a <type of user>	I want to <perform some task>	so that I can <achieve some goal>
1	Project manager	View a status report from each team member	Ensure the project stays on track.
2	Employee	Be reminded of upcoming deadlines	Complete my tasks on time.
3	Director	See the big picture view of department work	Stay in the loop.

[<https://www.smartsheet.com/>]

Who Uses User Stories?

An entire Agile team can use user stories for their work on a project, but here is a list of the key team members:

Product Owners: Ensure the delivery of a product that meets user needs.

Developers: Guide the work of the team.

Testers: Verify that the product performs as expected.

Technical Writers: Ensure that any support materials cover important use cases.

With the exception of developers, each of these people can act as a customer proxy, placing themselves in the role of a customer or user.

[<https://www.smartsheet.com/user-story-templates>]

What Are Some Benefits of User Stories?

- It's a simple way to see what new features and capabilities are needed.
- They clarify the functionality needed to solve customer problems.
- They are easy to understand and remember.
- They focus on business value and customer needs.
- They make prioritization easy.
- They focus on how potential customers will use the product.
- They can save time, as there are fewer false starts.
- They can be used to trace the history of the product by tracking what features were added in each iteration.
- They shift the focus from writing requirements to talking about them.
- They can have different levels of detail.
- By breaking down work into chunks, they provide flexibility in implementation.
- Technical specs are left to the developers.
- They drive collaboration and creative solutions.
- They improve ROI and team morale.

[<https://www.smartsheet.com/user-story-templates>]

SMART Tasks (user stories to PB tasks)

[<https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>]

There is an acronym for creating effective goals: "SMART"

S – Specific

M – Measurable

A – Achievable

R – Relevant

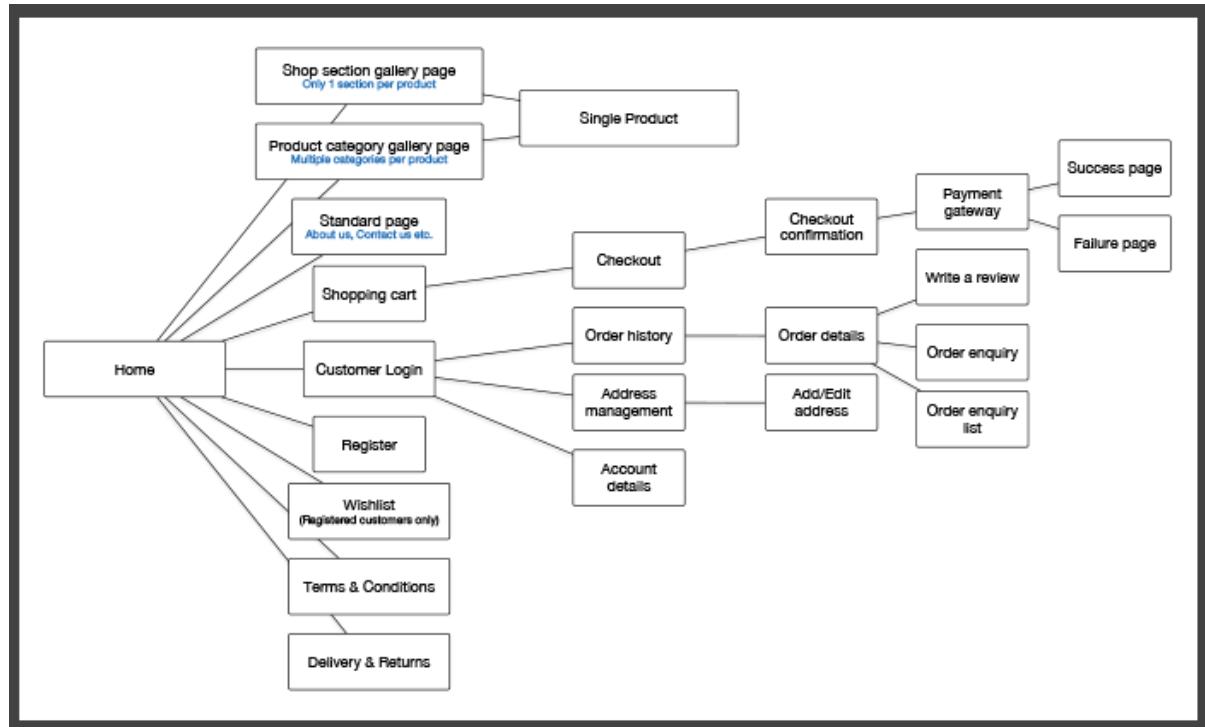
T – Time-boxed.

(There are a lot of variations in what the letters stand for.) These are good characteristics for tasks as well.

Navigation diagram

Web shop navigation

Exit/logout ?
Perhaps possible
from every view ?



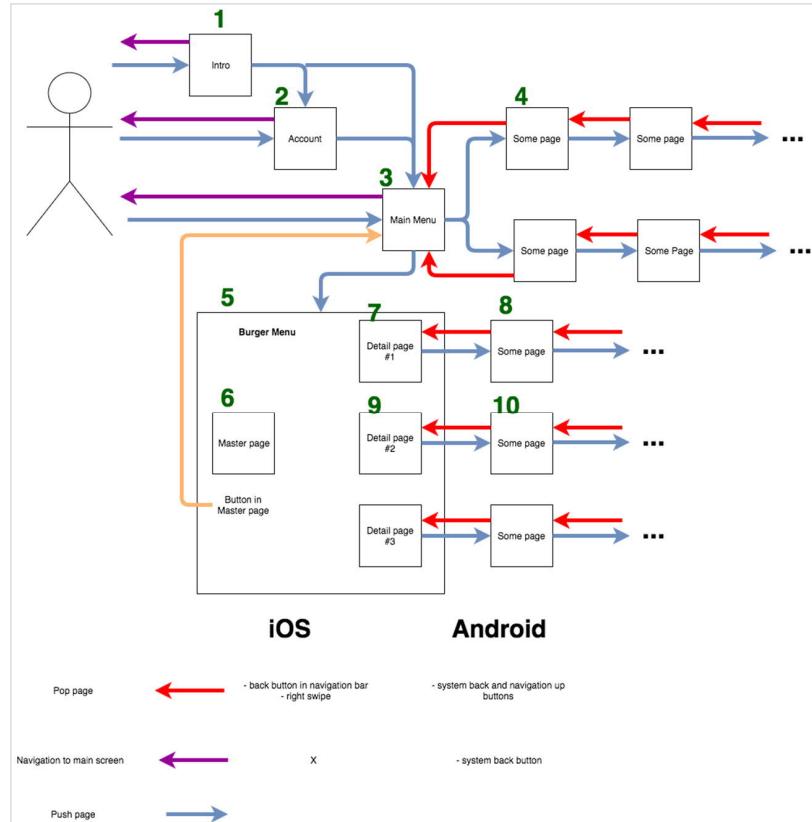
[www.nexusdp.co.uk/blog/2016/04/8-step-guide-to-creating-a-successful-small-business-website/]

22.09.2020 14.02

TUNI / COMP.SE.100-EN

90

Navigation diagram



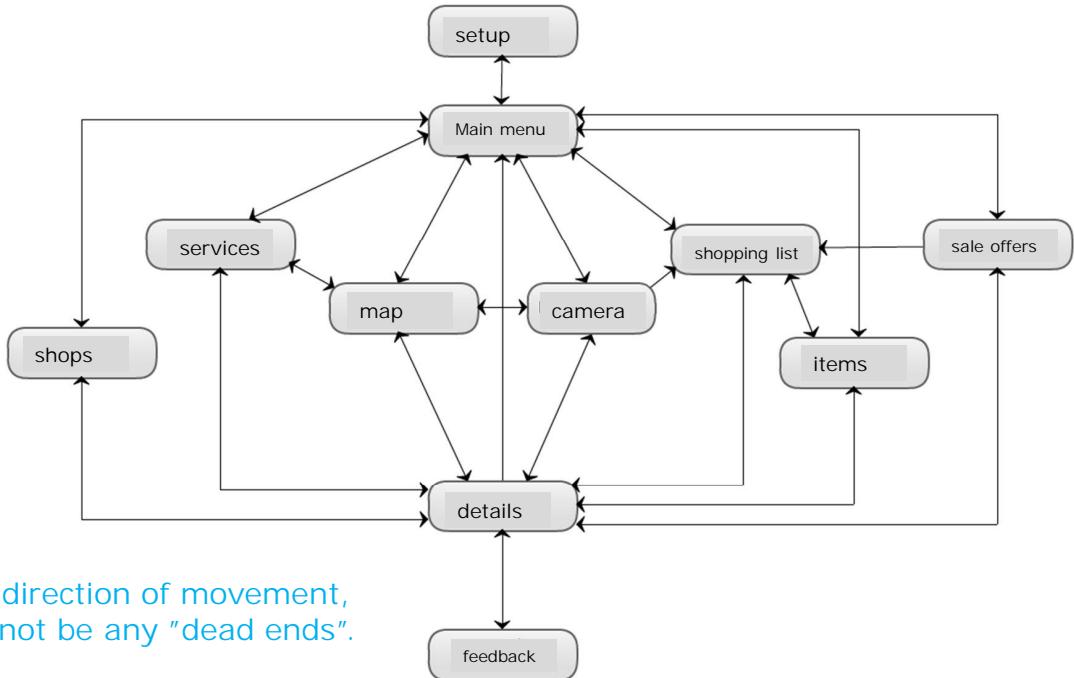
[i.stack.imgur.com/GF5nq.png]

22.09.2020 14.02

TUNI / COMP.SE.100-EN

91

Navigation diagram / chart / map



22.09.2020 14.02

TUNI / COMP.SE.100-EN

[online diagramming & design] [creately.com](#)

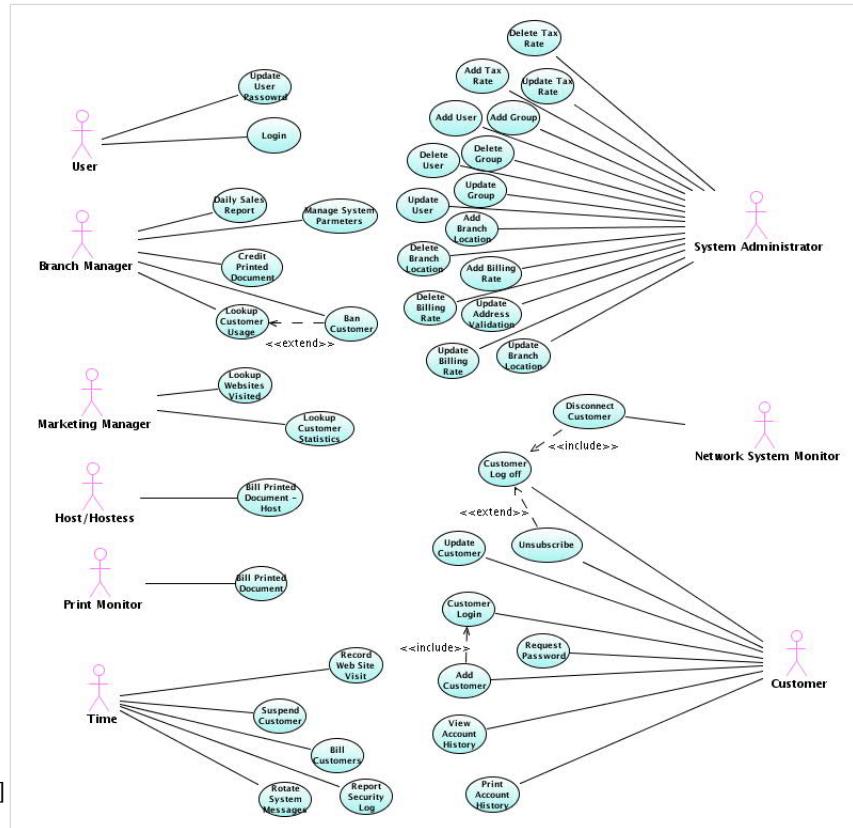
92

Many examples exist, e.g.

www.uml-diagrams.org/

See also Additional material at Moodle.

Use Case diagram (International internet cafe)



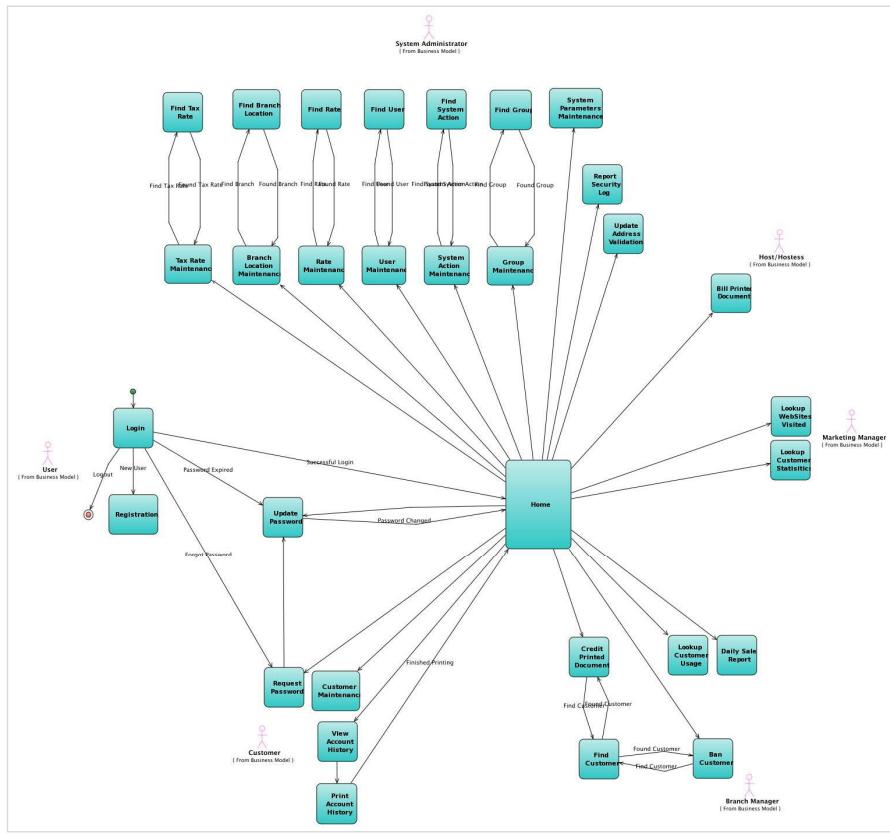
[cafesource.sourceforge.net/phase1/]

TUNI / COMP.SE.100-EN

22.09.2020 14.02

94

Navigation diagram (International internet cafe)



[cafesource.sourceforge.net/phase1/]

TUNI / COMP.SE.100-EN

22.09.2020 14.02

95



about UML and diagrams in general

Question: are UML diagrams updated during an agile software project ?

Answer: well, it depends...

- at requirements phase, yes, until requirements are done
- during implementation, if needed
- diagrams may be used to write system test cases
- maintenance people like up-to-date digrams !

Remember always update code comments, when making modifications and bug fixes (for maintenance phase).

Highlights - What to remember

- Diagrams are needed to help at requirements gathering and specification phase. No just diagrams nor just text describes the system well; both are needed together supporting each other.
- context diagrams, Use Case diagrams and navigation charts are the most common diagrams at requirements phase
- there may be some variations in actual use ("UML-BUT"), as long as all parties/stakeholders understand diagrams
- remember also to make a data dictionary (= glossary)
- diagrams are here to help common understanding
- later at your work, use whatever kind and style of diagrams that help.

Now the additional L4 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

Now the additional L4 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

Now the additional L4 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

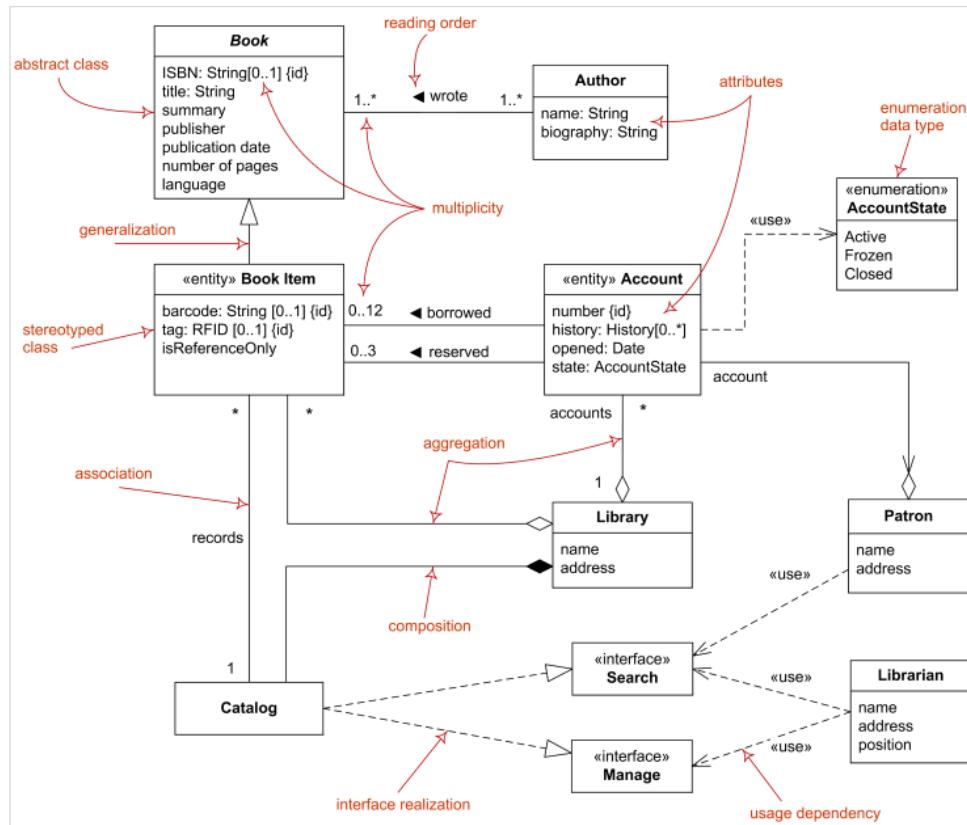
Now the additional L4 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

Entity / context diagram

Not all the details which are in this example are needed in your PA.

[www.uml-diagrams.org]



TUNI / COMP.SE.100-EN

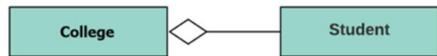
23.09.2020 12.23

104

<https://www.guru99.com/uml-class-diagram.html>

Aggregation

Aggregation is a special type of association that models a whole- part relationship between aggregate and its parts.



For example, the class **college** is made up of one or more student. In aggregation, the contained classes are never totally dependent on the lifecycle of the container. Here, the college class will remain even if the student is not available.

Composition:



The composition is a special type of aggregation which denotes strong ownership between two classes when one class is a part of another class.

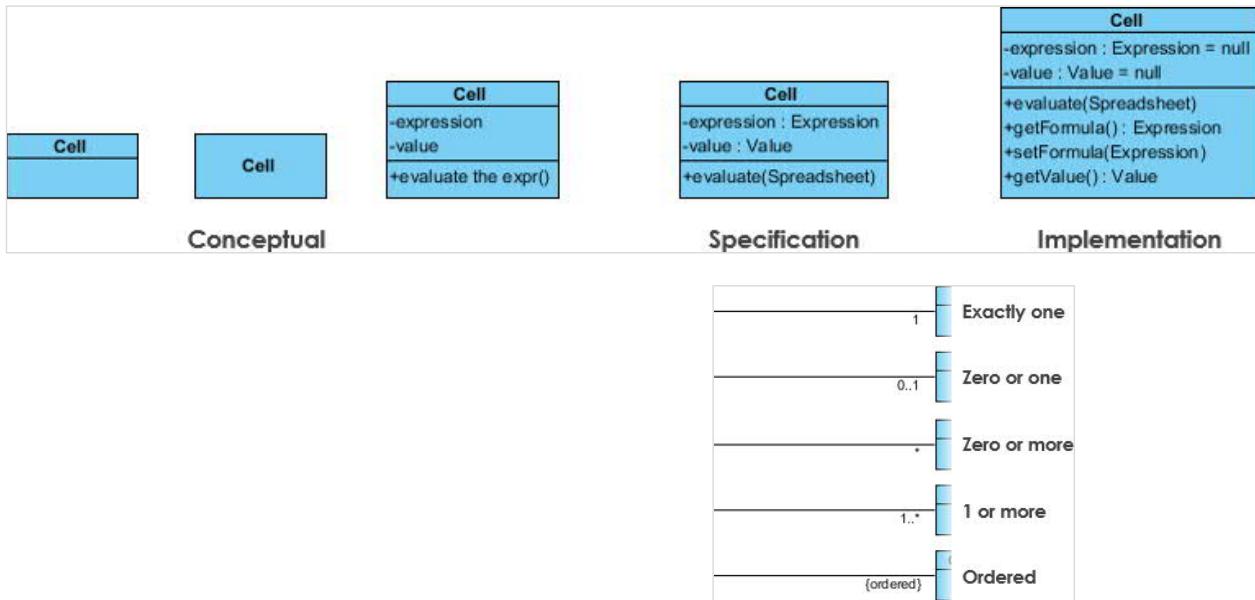
For example, if college is composed of classes student. The college could contain many students, while each student belongs to only one college. So, if college is not functioning all the students also removed.

TUNI / COMP.SE.100-EN

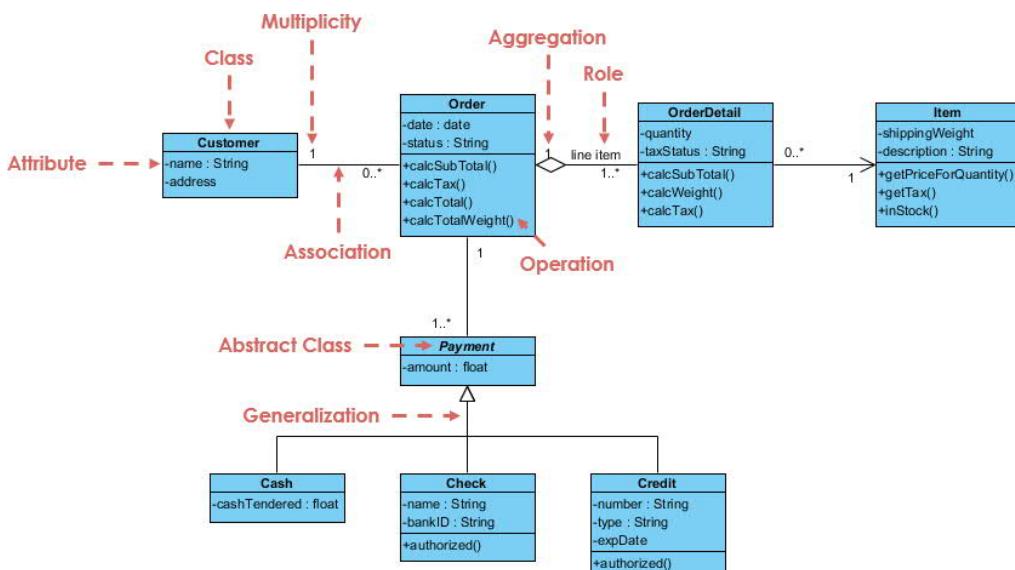
22.09.2020 14.02

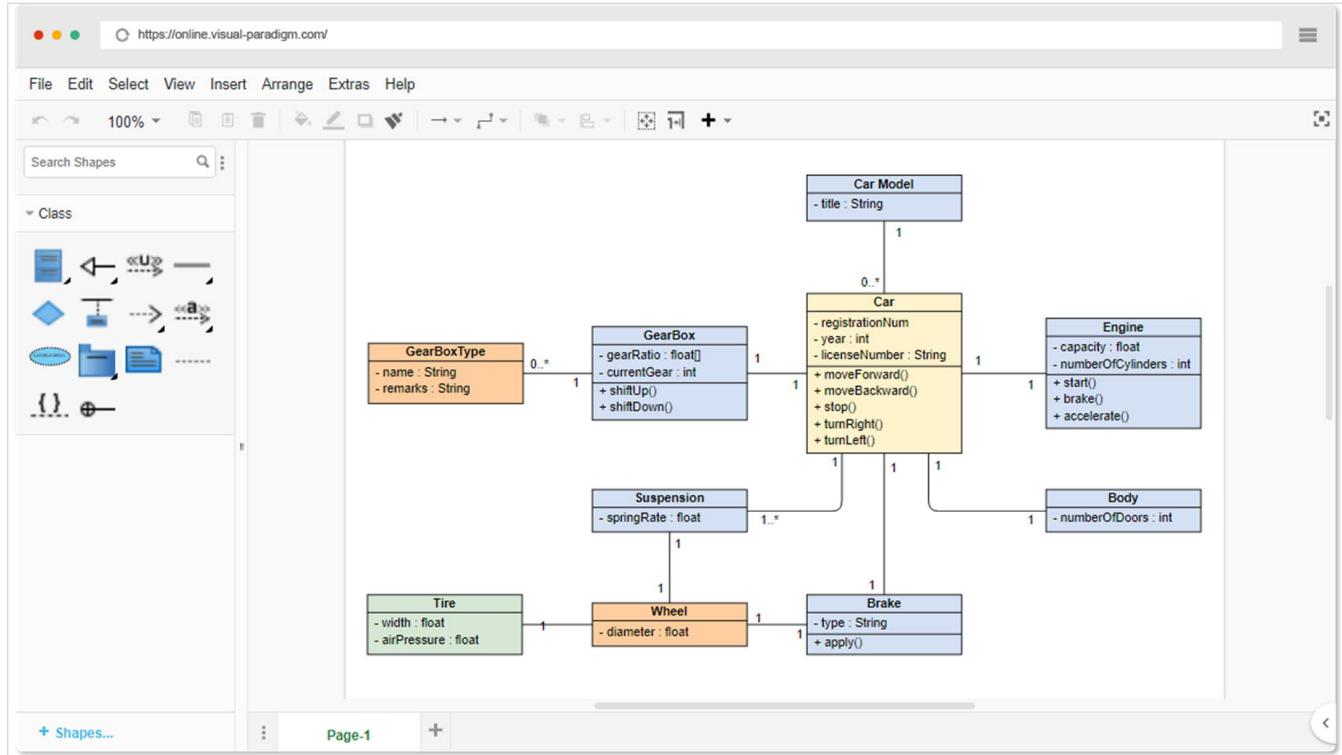
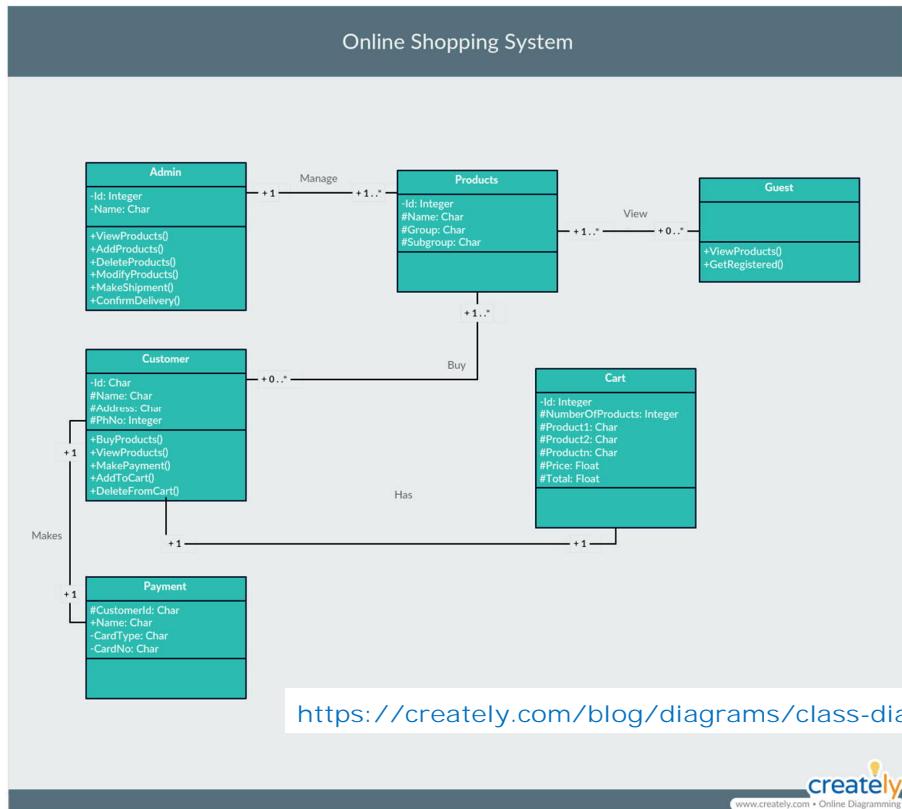
105

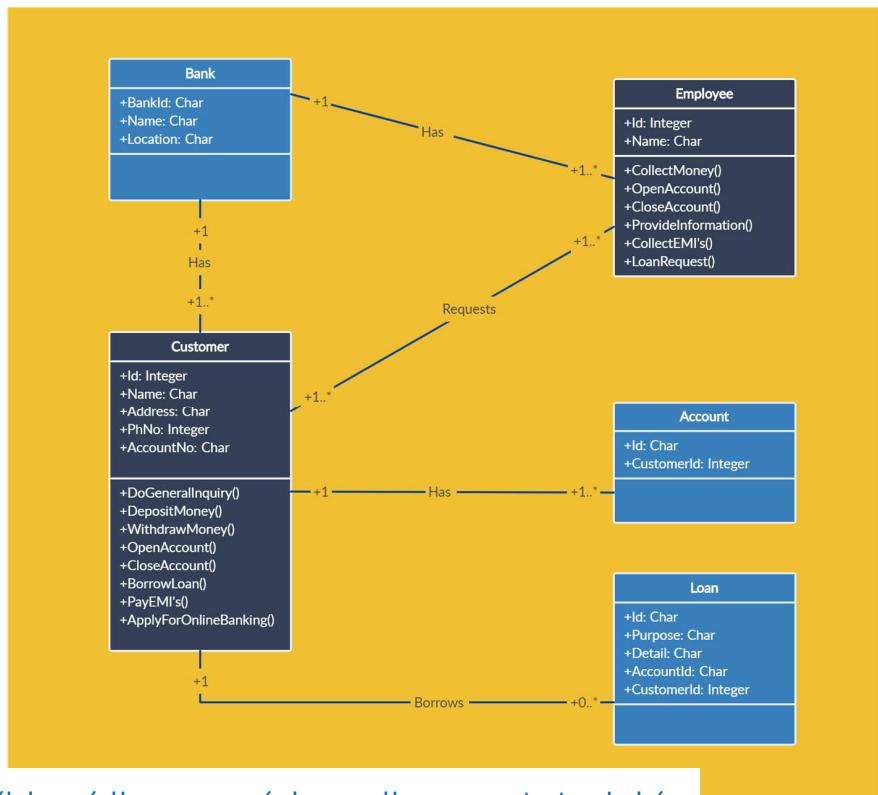
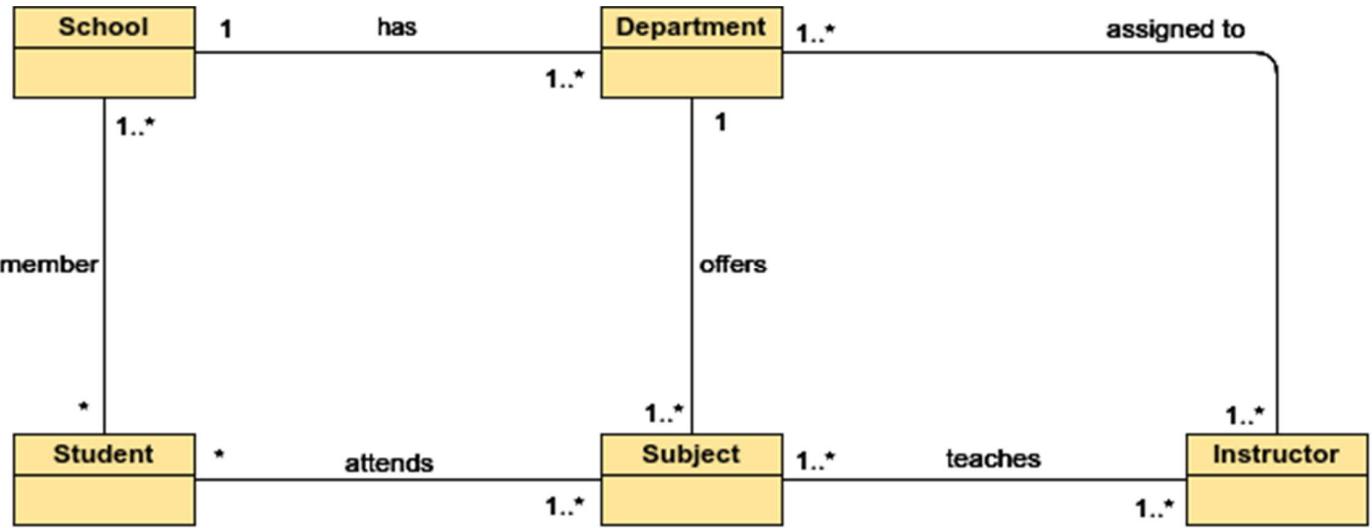
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

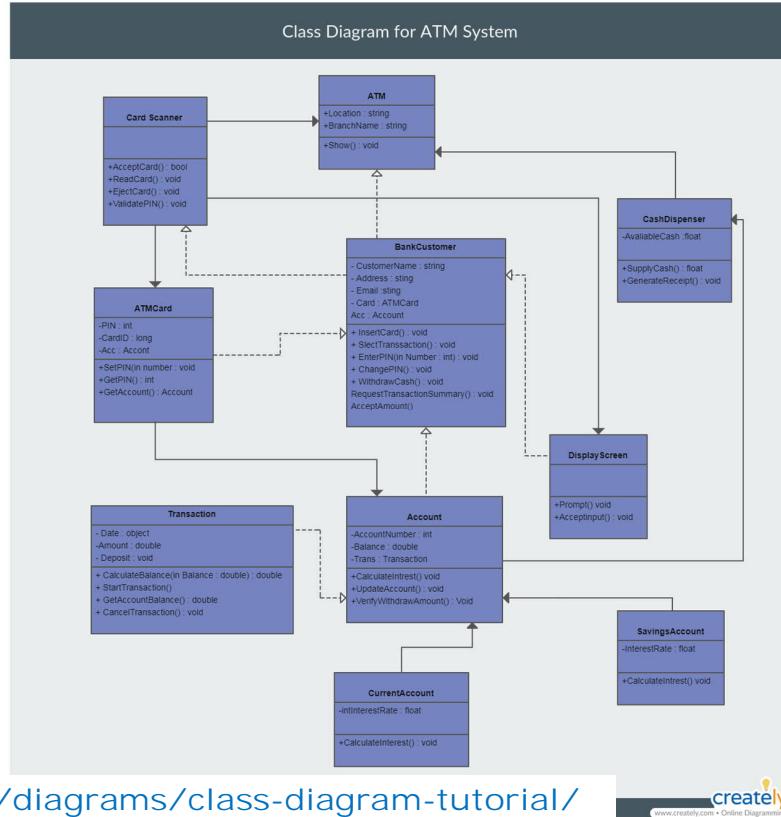


<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>





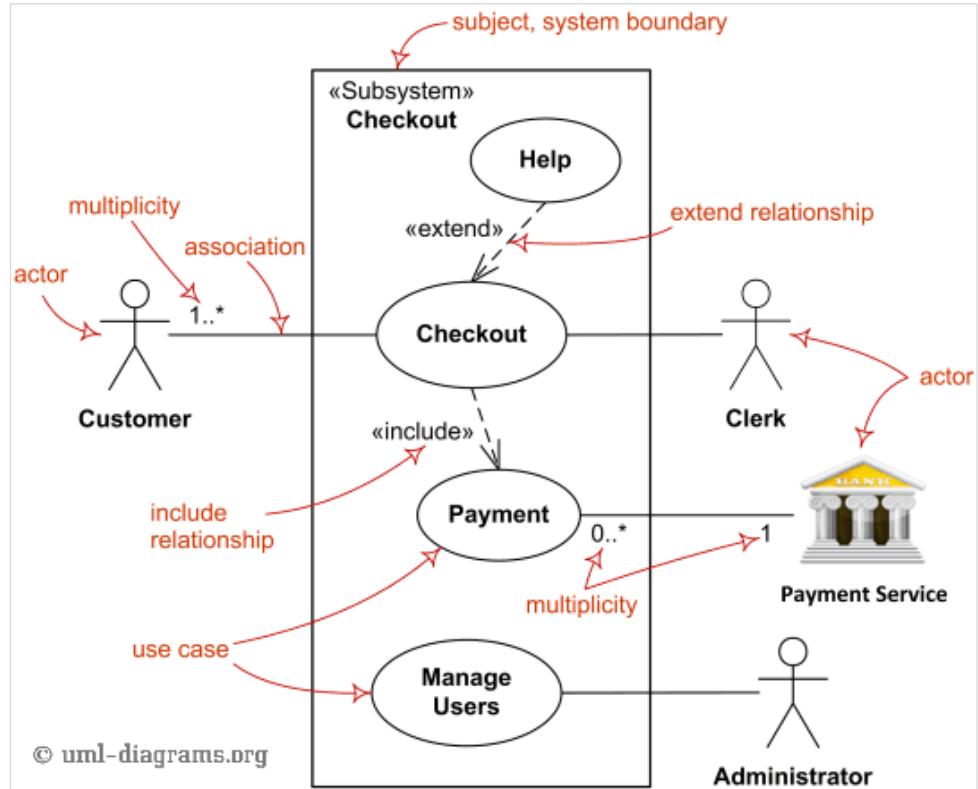




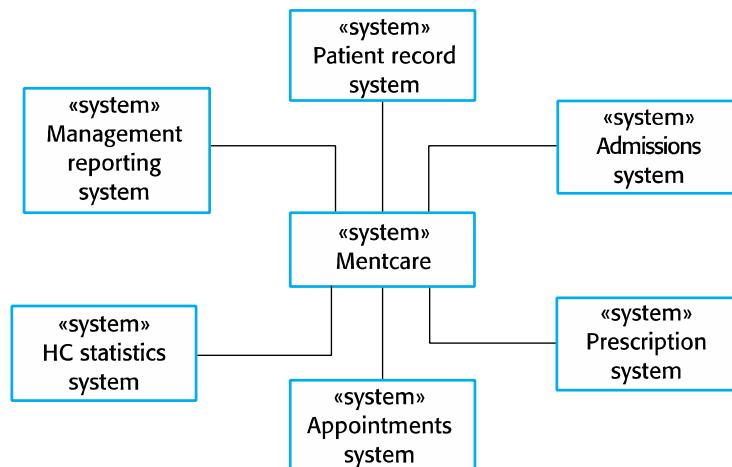
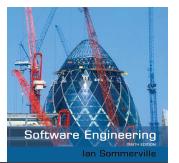
<https://creately.com/blog/diagrams/class-diagram-tutorial/>

Use case diagram

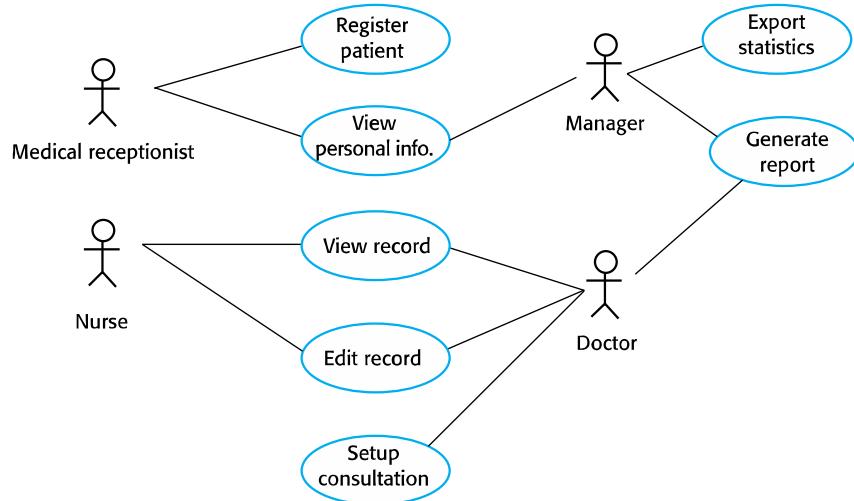
Do not use extend or include, if you are not absolutely sure what those mean.



The context of the Mentcare system



Use cases for the Mentcare system

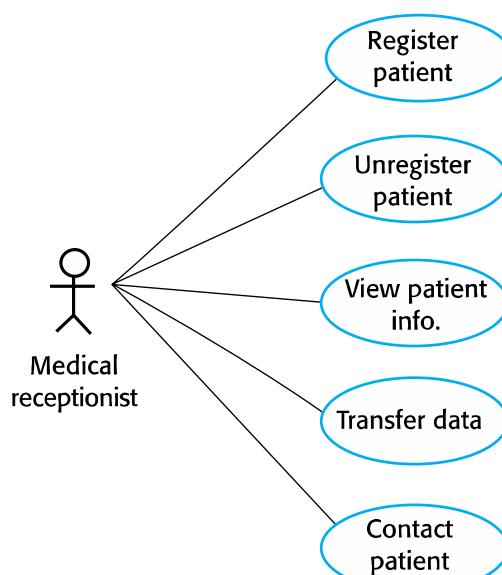
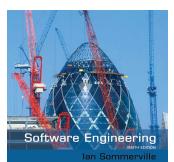


30/10/2014

Chapter 4 Requirements Engineering

116

Use cases in the Mentcare system involving the role 'Medical Receptionist'

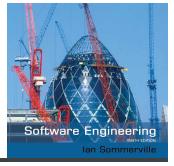


30/10/2014

Chapter 5 System Modeling

117

Use cases



- ✧ Use-cases are a kind of scenario that are included in the UML.
- ✧ Use cases identify the actors in an interaction and which describe the interaction itself.
- ✧ A set of use cases should describe all possible interactions with the system.
- ✧ High-level graphical model supplemented by more detailed tabular description (see Chapter 5).
- ✧ UML sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system.

30/10/2014

Chapter 4 Requirements Engineering

118

Use Case diagram and User Story (Use Case)

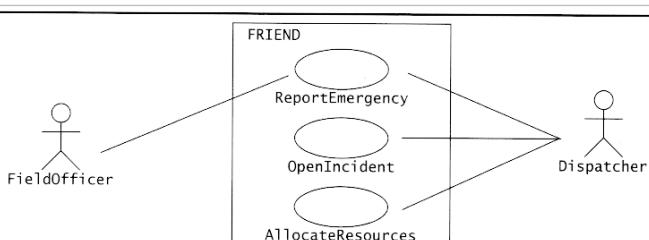
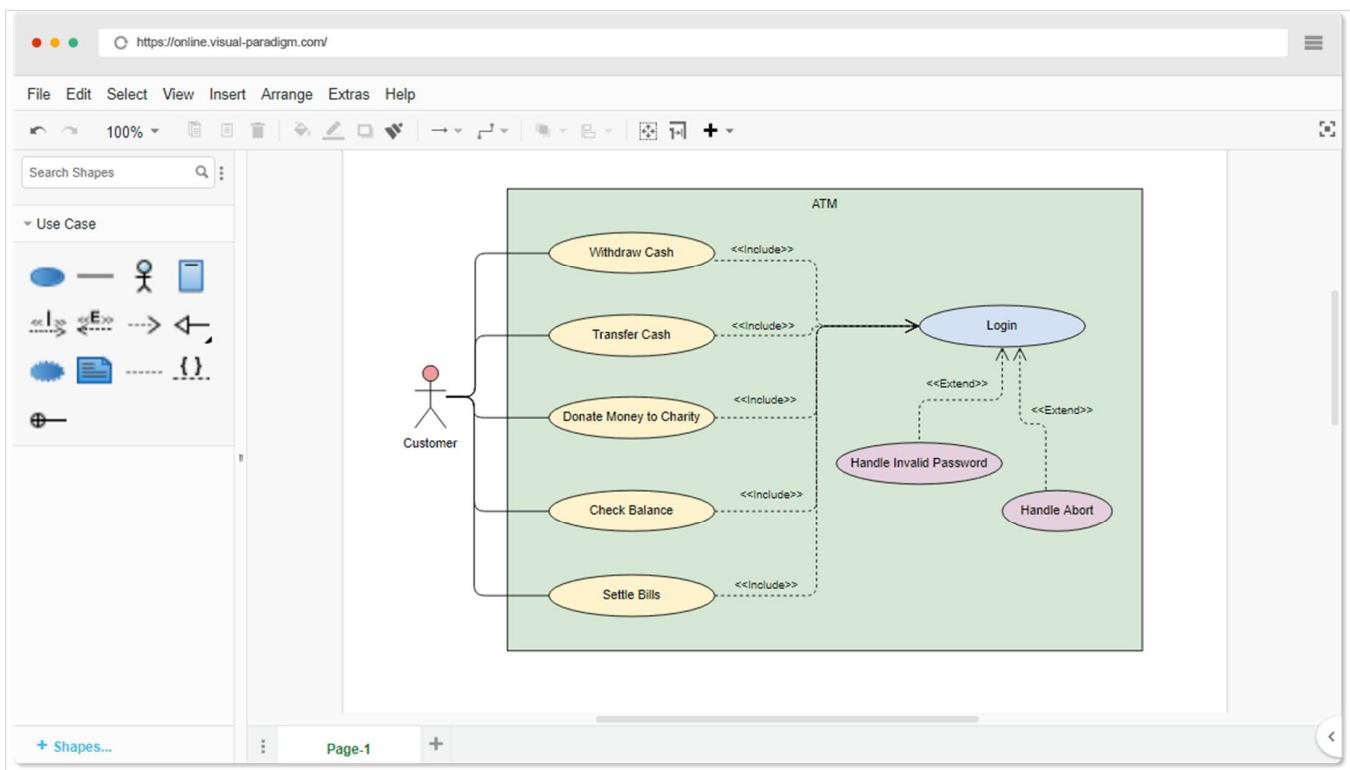


Figure 2-13 An example of a UML use case diagram for First Responder Interactive Emergency Navigational Database (FRIEND), an accident management system. Associations between actors and use cases denote information flows. These associations are bidirectional: they can represent the actor initiating a use case (FieldOfficer initiates ReportEmergency) or a use case providing information to an actor (ReportEmergency notifies Dispatcher). The box around the use cases represents the system boundary.

Once again, terminology may vary:
Start state = entry = stimulus
End state = exit = response.

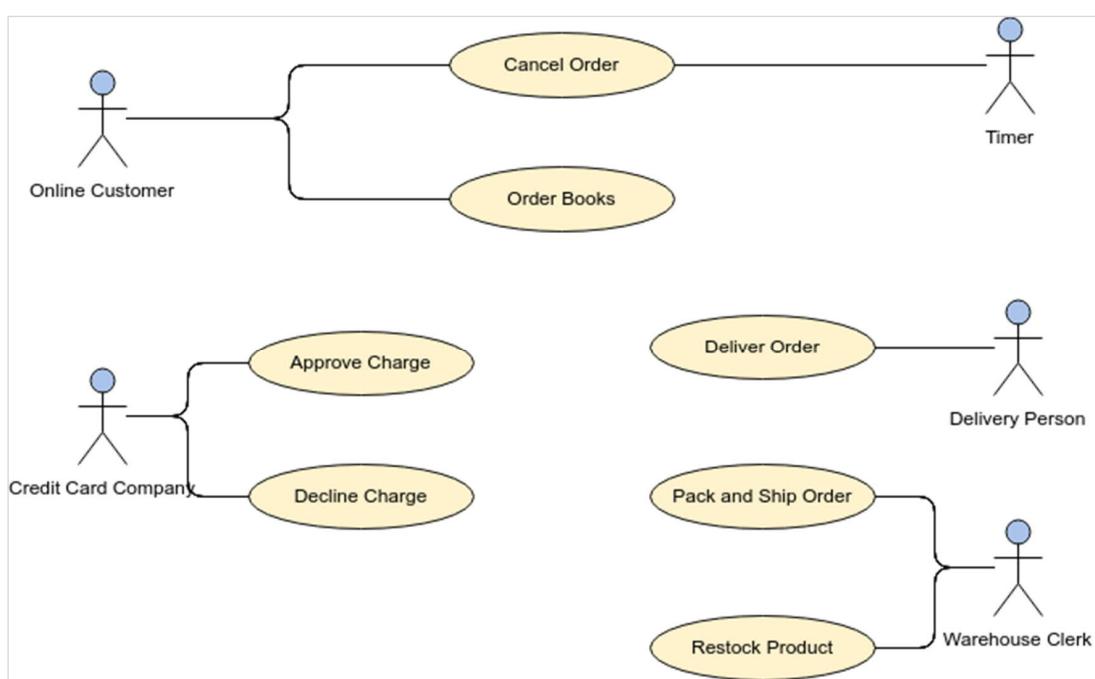
<i>Use case name</i>	ReportEmergency
<i>Participating actors</i>	Initiated by FieldOfficer Communicates with Dispatcher
<i>Flow of events</i>	<ol style="list-style-type: none">1. The FieldOfficer activates the “Report Emergency” function of her terminal.2. FRIEND responds by presenting a form to the FieldOfficer.3. The FieldOfficer fills out the form by selecting the emergency level, type, location, and brief description of the situation. The FieldOfficer also describes possible responses to the emergency situation. Once the form is completed, the FieldOfficer submits the form.4. FRIEND receives the form and notifies the Dispatcher.5. The Dispatcher reviews the submitted information and creates an Incident in the database by invoking the OpenIncident use case. The Dispatcher selects a response and acknowledges the report.6. FRIEND displays the acknowledgment and the selected response to the FieldOfficer.
<i>Entry condition</i>	<ul style="list-style-type: none">• The FieldOfficer is logged into FRIEND.
<i>Exit condition</i>	<ul style="list-style-type: none">• The FieldOfficer has received an acknowledgment and the selected response from the Dispatcher, OR• The FieldOfficer has received an explanation indicating why the transaction could not be processed.
<i>Quality requirements</i>	<ul style="list-style-type: none">• The FieldOfficer’s report is acknowledged within 30 seconds.• The selected response arrives no later than 30 seconds after it is sent by the Dispatcher.

Figure 2-14 An example of a use case, ReportEmergency.

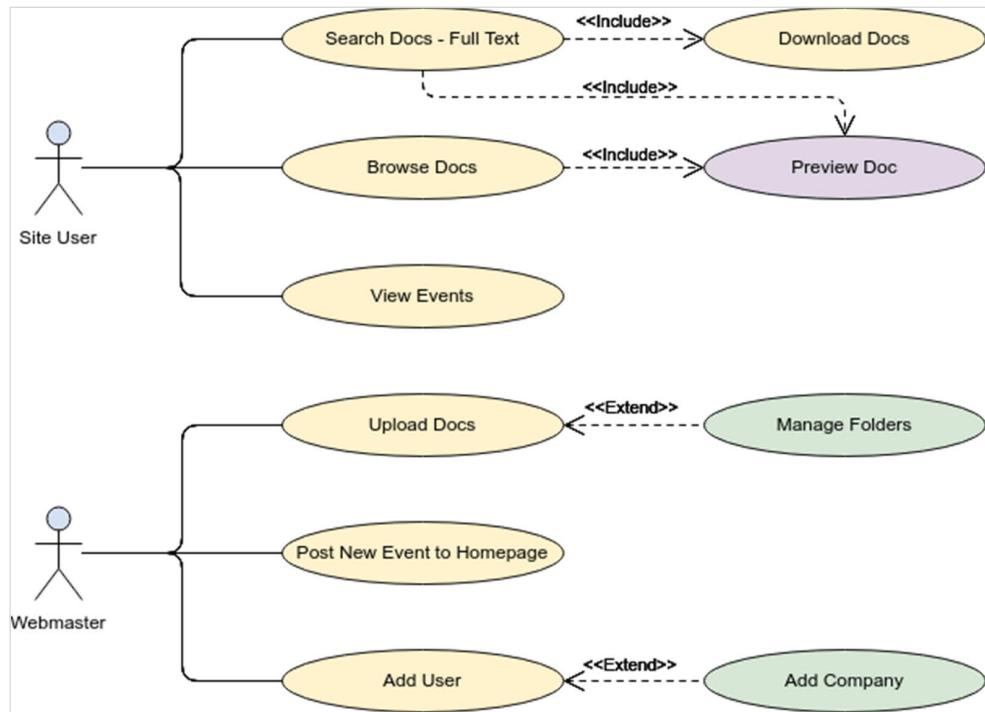


<https://online.visual-paradigm.com/diagrams/templates/use-case-diagram/>

Order process system

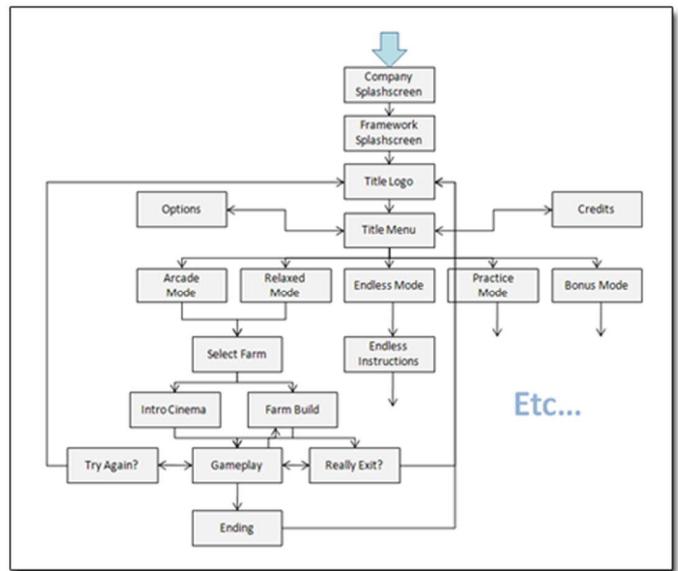
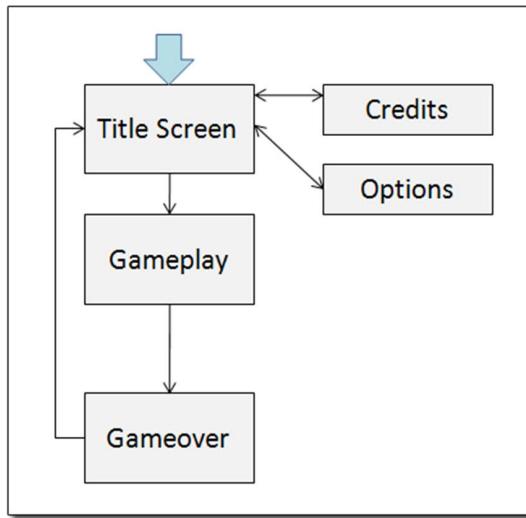


Website



Navigation diagram

Navigation diagram / map / grid / chart (feature tree)



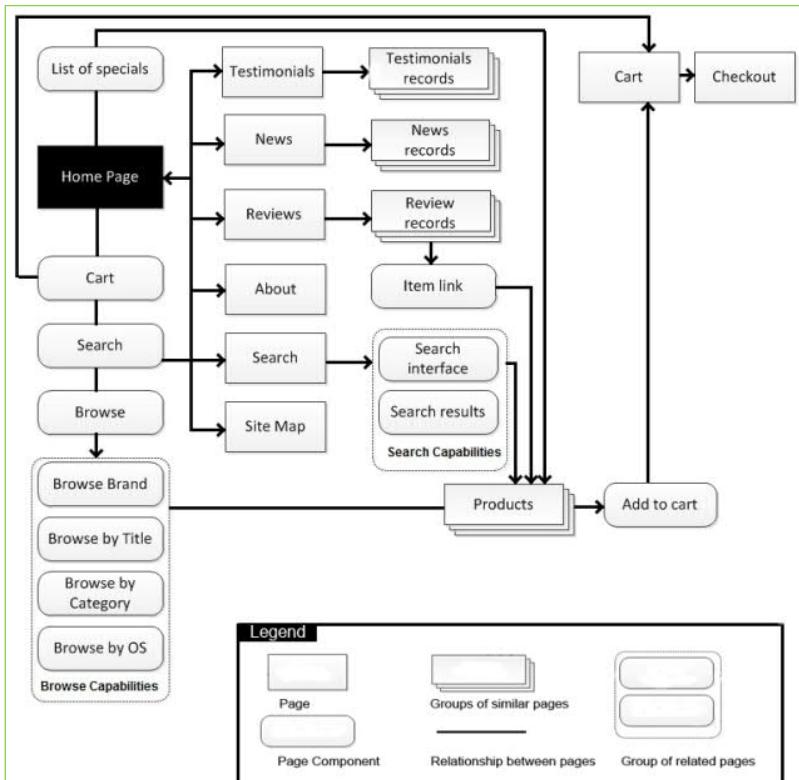
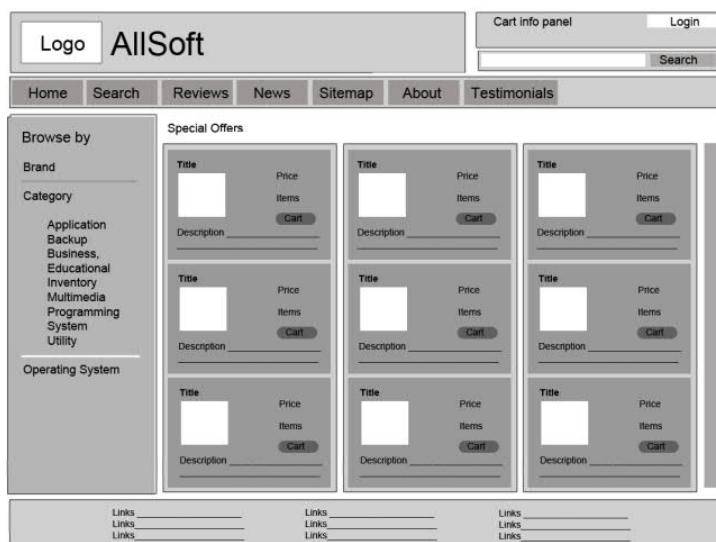
[rivermanmedia.com/gui-design-tip-the-navigation-grid/]

23.09.2020 12.25

TUNI / COMP.SE.100-EN

124

GUI proto and navigation



[s2713275.wordpress.com/2012/09/28/part-3-allsofts-web-site-design-wire-frames/]

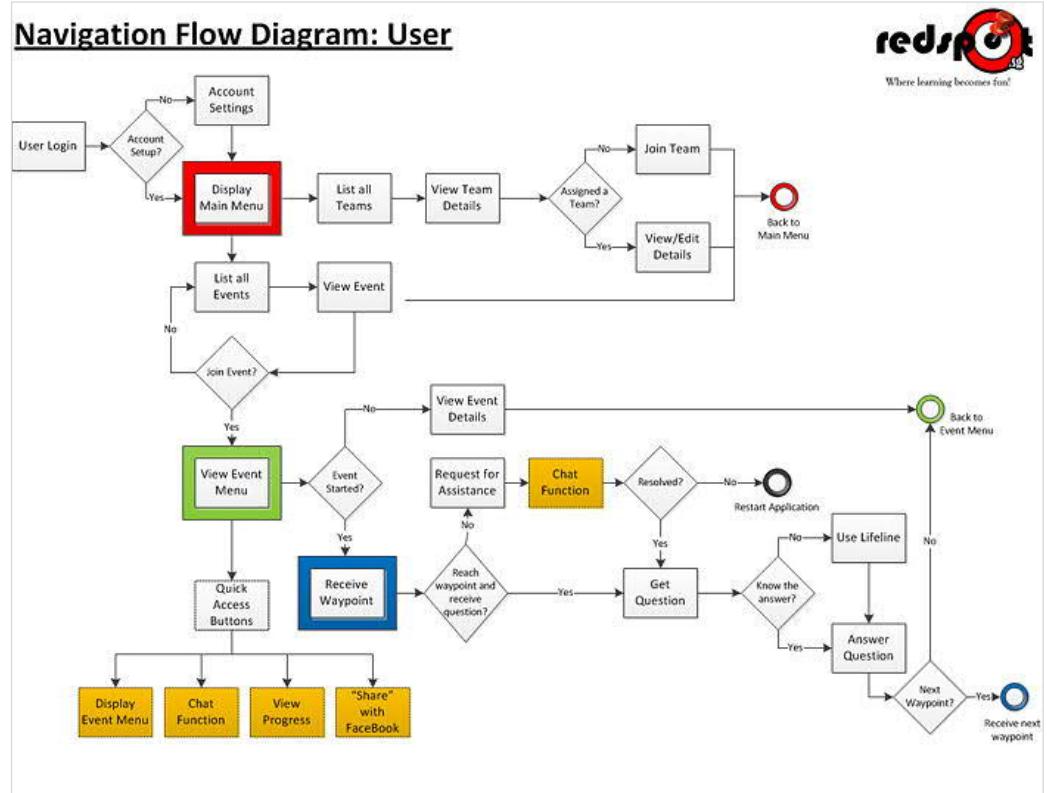
23.09.2020 12.25

TUNI / COMP.SE.100-EN

125

Navigation with Yes/No diamonds

[wiki.smu.edu.sg/is480/...]



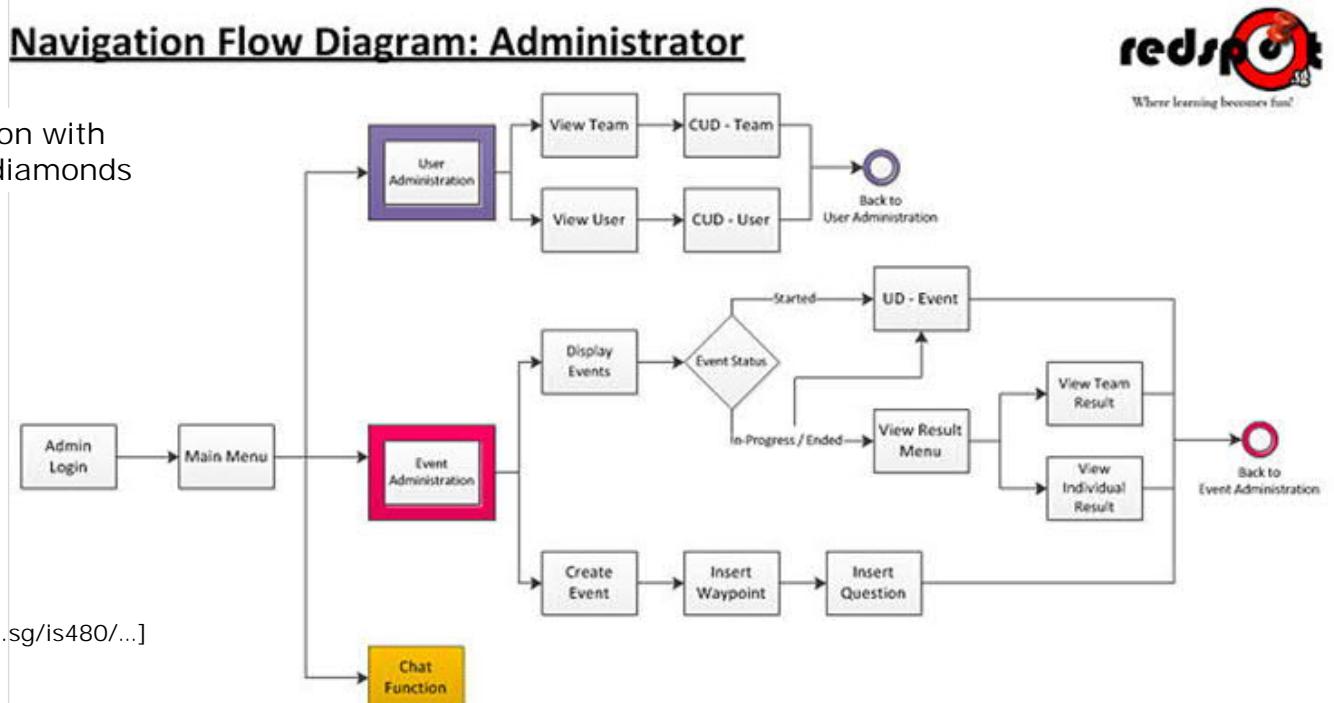
23.09.2020 12.26

TUNI / COMP.SE.100-EN

126

Navigation with Yes/No diamonds

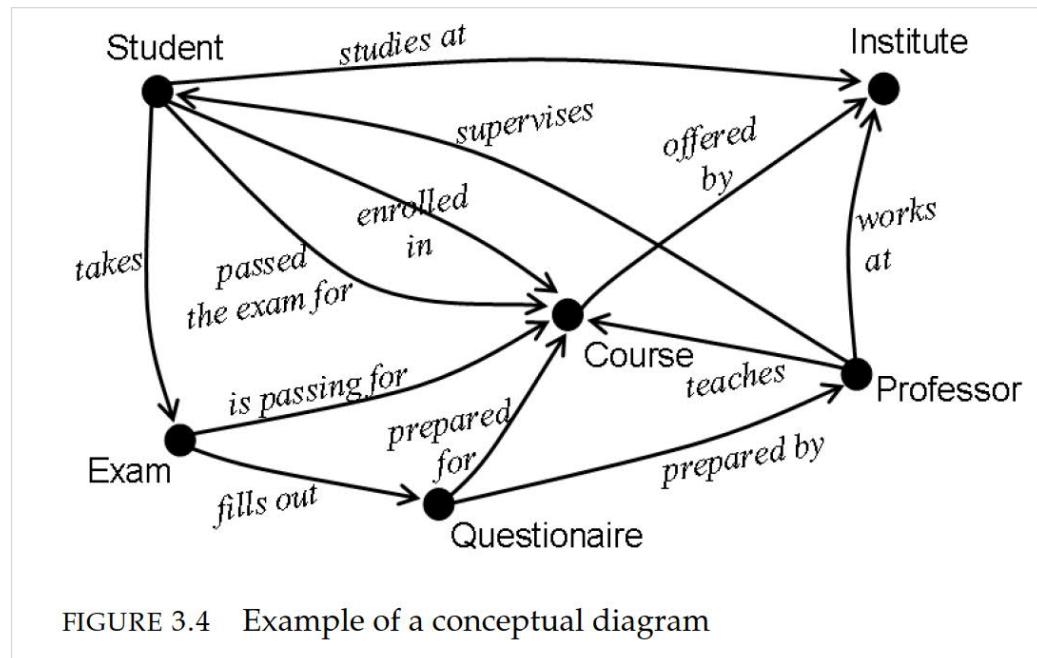
[wiki.smu.edu.sg/is480/...]



23.09.2020 12.26

TUNI / COMP.SE.100-EN

127

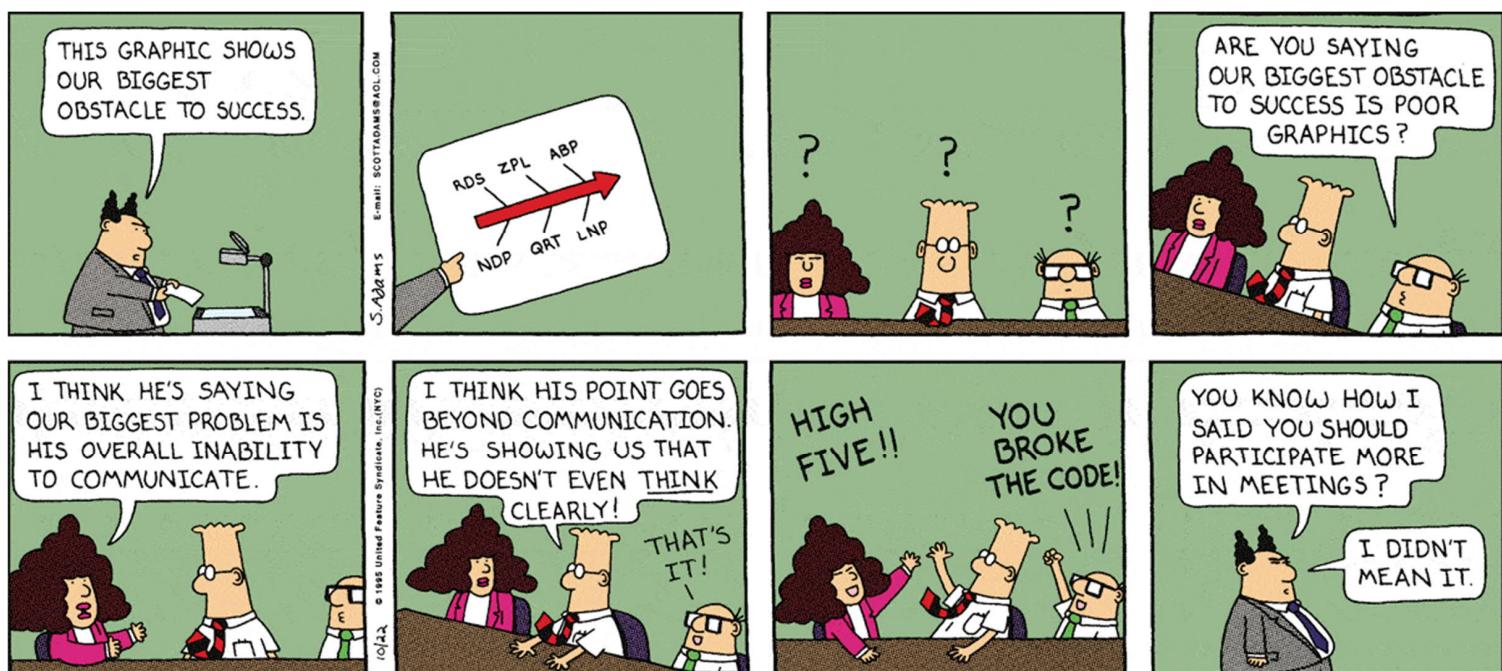
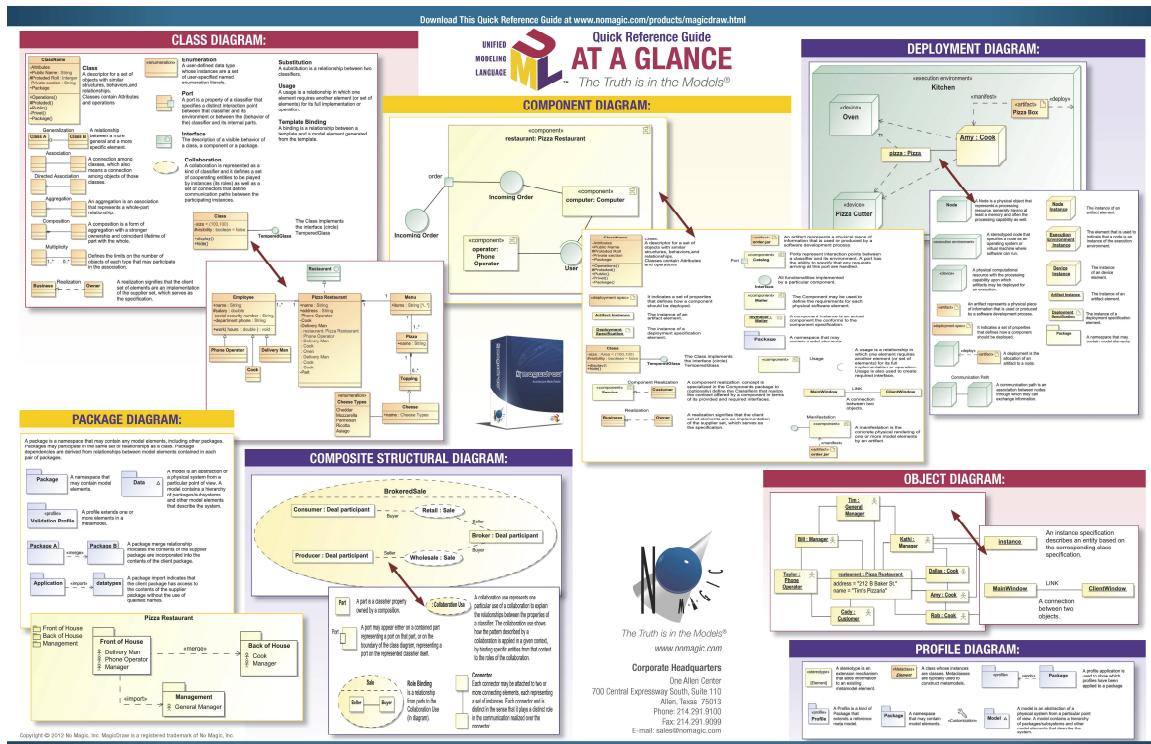


[Rule-Based Design, 2013]

<https://www.uml-diagrams.org/>

<https://creately.com/diagram/example>

There are many UML reference cards and cheat sheets available



Now the additional L3
extra slides set ends here

Now the additional L3
extra slides set ends here

22.09.2020 14.02

TUNI * COMP.SE.100-EN

134